



Release Notes v5

What's New in Version 5.0

A **What's New** help item is available for highlighting the many new features in TableCurve 2D release 5. This item is available in the main Help menu of the program.

Tutorials

The **tutorials** for this release have been increased to eight and can be accessed directly from the Help menu. Six of these tutorials are completely new and offer valuable information in using many of the data preconditioning procedures effectively.

On-Line Help and PDF Documents

The documentation is now entirely electronic. All of the information previously in printed form, as well as a large measure of additional documentation, is available in the on-line help. There are now full color graphs that illustrate the influence of key parameters in the **Non-Linear Equations**. In addition to curve-fitting information, you will find a good summary of Fourier analysis, Eigendecomposition, and Autoregressive modeling.

The tutorials, the linear and non-linear equations descriptions, and the full on-line reference system have been formatted as letter-size, full color Adobe Acrobat PDF electronic documents. If desired, you can print or view these reference documents using the Acrobat reader included on the distribution CD. These files are installed into the PDF subdirectory of the TableCurve 2D v5 directory. The PDF subdirectory can be deleted if PDF files are not of value to you.

What's New in v5

- 1. State-of-the-Art Numeric Prediction
- 2. Automation
- 3. MS Excel and MS Word Enhancements
- 4. Derivative Enhancements
- 5. Denoising and Smoothing Enhancements
- 6. Reconstruction/Isolation of Signal Components
- 7. Improved Interpolation and Non-Parametric Estimation
- 8. Improved Evaluation Procedure
- 9. Multiple Curve-Fit References in a Single Graph
- 10. Sort by Maximum Residual
- 11. RTF Windows
- 12. Mark and Refit
- 13. Data Table Reset
- 14. Improved Residuals
- 15. New Connect Options in Graphs
- 16. Differencing, Cumulatives, Normalizations
- 17. Generating Data
- 18. Importing Equation References as Data Sources
- 19. More UDF Fn#s
- 20. XY Sectioning
- 21. Preferences
- 22. Additional Mouse Functionality
- 23. Automatic Window Placement in Review
- 24. Revisions to Toolbars and Menus
- 25. On-Line Documentation
- 26. Key Features and Enhancements Added to v4

1. State-of-the-Art Numeric Prediction

Unless a theoretically appropriate model is being fitted, extrapolations or predictions using TC2D model equations are often of limited utility. TC2D v5 offers state of the art AR (Autoregressive) procedures that offer the means to effectively forward or backward predict many data sets, including those with appreciable levels of noise.



The **AR Modeling and Prediction** procedure in the **Estimate** menu offers effective **autoregressive** forecasting and extrapolation. The AR algorithms include SVD (singular value decomposition) procedures for in-place noise removal. Multiple orders can be simultaneously plotted and stabilizations are available for roots that lie outside the unit circle. The points that are to be processed can be specified, allowing predictions based on a data segment to be compared with subsequent data. The extent of the prediction is variable and white noise can be added to test the robustness of an algorithm's prediction.

One of the weaknesses of traditional AR prediction methods is their susceptibility to noise. Generally, even a small measure of random noise is sufficient to render predictions unreliable or meaningless. TC2D v5 solves this problem by offering nine different procedures (three to predict ahead, three to predict earlier data, and three that predict in both directions). Of the three major least-squares algorithms included in v5, two offer in-place noise removal using advanced SVD and eigendecomposition methods.

TC2D's Autoregression option offers the means to view **complex roots**, **order selection criteria**, and **singular values** in order to optimize the AR model order and other algorithm parameters. Residuals, numeric summaries, and both numeric and filter export are available. TC2D's AR procedure is also unique in that multiple AR orders can be simultaneously graphed and displayed.

AR filter code generation is supported in all of TC2D's supported languages. You can also save a filter to disk that is treated as a separate AR algorithm. It can be imported and applied to any data set desired.

2. Automation

While TC2D is designed to be interactive, there are those instances where a large number of data sets need to be processed using only one or two different models.



For these data sets, TC2D v5 offers an **integrated automation** that enables large numbers of data sets to be processed in an unattended fashion.

The primary automation uses MS Excel as the data source. If they are not already there, you simply place all your data sets in an Excel spreadsheet. These multiple sets are then processed with the ease of a single set. The automation output can be written to an MS Word (or generic RTF) file for all graphs and numeric summaries, and to MS Excel for numeric data. It is a simple matter to specify one sheet in the spreadsheet or the full file. Import from MS Excel 95, 97, and 2000 is supported. Export to MS Excel 95, 97, 2000 and MS Word 95, 97, and 2000 are supported. If you do not use MS Word, the exported reports can use a generic RTF format that can be read by any word processor, including Wordpad (ships with Windows).

Automation is integrated into each of TC2D's major procedures. It is thus possible to denoise 100 data sets in one unattended step, and then fit these sets to a target equation in a second unattended step. Since Excel is used for the output of the automation, these files can be used as the input to any other TC2D procedure or in other programs that support MS Excel.

TC2D's automation includes DLL support for those who want to write an external acquisition interface. TC2D v5 even includes a professional DLL automation interface for instrument manufacturers that want TC2D's functionality available within their own software.

3. MS Excel and MS Word Enhancements

In addition to supporting all releases of Excel through 2000 for data import, data can now be saved from any procedure in an Excel 95 or Excel 97/2000 format for import by TC2D, for use in Excel, or in some other application.



Most procedures offer a new **MS Word or RTF export**. The MS Word export is a streamed export that automatically opens MS Word and transfers the graphs and reports directly, all with a single button click. The graphs can be half or full page, and a number of controls are offered including landscape or portrait mode. In fact, the file-based (RTF) export allows you to export a mixture of landscape graphs and portrait reports, all in a single document.

4. Derivative Enhancements

An exceedingly small measure of noise can play havoc with derivatives. That is one reason why fitting an effective parametric model is a good way to achieve a smooth derivative. Through v4, however, TC2D offered only numeric derivatives for its 3491 linear equations and its 174 built-in non-linear equations. TC2D v5 now offers analytic derivatives, both first and second, for all built-in equations. Analytic Derivatives are set in the **Preferences** option.



Although Savitzky-Golay smoothing has been long a part of TC2D's automated smoothing, v5 now offers a separate **Savitzky-Golay Smoothing** procedure. This is probably the best non-parametric approach for derivatives. A much greater control of the algorithm is now offered. First through eighth derivatives are available, SG model orders 2 through 12 can be selected, the filter pass count is adjustable, and an option has been added to manage edge effects. The option now includes residuals and a numeric summary includes the smoothing coefficients.



For interpolation of a smooth derivative, there is also a new **Savitzky-Golay Spline Estimation** procedure. A constrained cubic spline is automatically used in this option to interpolate SG processed data.



The **Spline Estimation** and **Local Regression Estimation** options now offer derivatives through order 4 for all algorithms.

The separate Inspect Derivative options that were present in v4 have been removed.

5. Denoising and Smoothing Enhancements

While the Fourier filtering and the various smoothing procedures in v4 were effective in noise removal, TC2D v5 goes several steps further in offering state of the art denoising.



Completely new to v5 is **Eigendecomposition Denoising**. This is a non-parametric procedure where the highest ranked eigenmodes capture the deterministic component of data (signal) where the least-ranked eigenmodes contain the stochastic (noise) component. Unlike the Fourier filtering, where the separation is based on frequency content, this method separates based on the strength of the components. When data sizes allow a sufficiently high order eigendecomposition, it is possible to filter virtually all noise from a given data set without distorting the underlying signal. This is the ultimate in noise removal.



TC2D v5 now includes a separate **Fourier Denoising** procedure. This is a simplified Fourier domain editing option designed exclusively for one-click noise removal. New to v5 is the option to apply a **data taper** and to filter based on either the magnitude of the FFT or the frequency thresholding previously offered. **Exact n FFTs** are now used to maximize the accuracy of the transform. Export now includes a much expanded FFT data summary and a new **FFT numeric summary** that includes a linear multi-sinusoidal fit that further optimizes the spectral information in the FFT. Up to 25 sine components are automatically least-squares fitted. The FFT can now be displayed in a variety of power, amplitude, dB, and other formats.



The **Automated Smoothing** procedure has been updated. The Eigendecomposition procedure has been added as well as an effective Kaiser-Bessel time-domain smoothing filter algorithm. Newly integrated is automatic spline interpolation for algorithms that require uniformly spaced data. Derivatives are now available for all of the smoothing procedures and are based upon a B-spline interpolant.

6. Reconstruction/Isolation of Signal Components



Eigenfiltering is useful for much more than noise reduction. Often a data set is a complicated mixture of a number of individual components that arise from a variety of sources. TC2D v5 includes a new **Eigendecomposition Filtering** procedure that makes it possible to isolate these discrete elements within a data stream. The separation is non-parametric and is often quite effective. While it is often most successful with oscillatory components of differing powers, mixtures of exponential decays can sometimes be effectively separated in this way.



When components have close to equal power, but separate spectral content, a Fourier domain filtration can often isolate the components. TC2D v5 offers an improved **Fourier Filtering** procedure. This is a full-power Fourier domain editor that represents a significant enhancement over the functionality in v4. **Exact n FFTs**, a selection of **data tapers** useful for reconstruction, and full graphical isolation of components have been added.

7. Improved Interpolation and Non-Parametric Estimation



The **Spline Estimation** option now includes a user-defined least-squares B-spline fitting procedure (the knot sequence is imported from any supported file format).



For data with oscillatory properties, TC2D v5 offers a **Fourier Estimation** procedure that directly evaluates the individual Fourier components. Any data range and density can be specified.



The local regression procedure has been revised. The **Local Regression Spline Estimation** procedure now uses spline interpolation to insure continuous and smooth derivatives.

Whereas the Evaluation procedure was needed in v4 for all interpolations, TC2D v5 now offers the means to directly specify and plot interpolated output. For this reason, the two Digital Filter options present in v4 have been removed. The **Spline Estimation** and **Local Regression Spline Estimation** now offer this same functionality.

8. Improved Evaluation Procedure



The **Evaluation** option now uses a single sorted list window. The columns can be sorted by x or y values, or by the magnitude of confidence interval. The numeric precision is adjustable. All roots are now done in the table generation procedure and the Direct to File option now offers all file formats. It is possible to generate a table by specifying either the sample increment or count; previous values are saved across sessions. Evaluation export is now spreadsheet-tailored to manage the different kinds of evaluations.

You can now save and recall evaluation sequences. Further, evaluation sequences are now automatically updated whenever an equation or algorithm setting changes.

9. Multiple Curve-Fit References in a Single Graph



In the Curve-Fit review, it is now possible to specify up to four **Equation References** in a given curve-fit plot. These can be other equations in the current fit, or equations saved to disk from prior fits. This will enable more extensive graphical comparisons between the various equations.



For comparing data rather than equations, the **Compare with Reference** option has been extended to also allow up to four data references to be compared with the current data set. References can now be from any data source. These can also include curve-fit references (equations saved to disk). For an equation reference, discrete data is generated.

10. Sort by Maximum Residual

For certain applications, the desired equation is one that minimizes the maximum error. To make this type of fitting easy, TC2D v5 includes a **Sort by Maximum Error** option for the Equation List. The maximum absolute error has also been added to the **Numeric Summary**.

11. RTF Windows

RTF Text windows can now be specified. This option is set in the [Preferences](#). With RTF windows, you can now highlight and copy any part of a report for pasting into a different application. The faster display windows used in v4 are also available and are automatically used for large table sizes.

12. Mark and Refit



In v4, latent point processing made it possible to tag points (usually outliers) in the Review that would be rendered inactive when returning to the main menu. The data could then be refitted absent the influence of these points. This has been replaced in v5 by a direct [Mark and Refit](#) option. This option makes it possible to exclude points based outside a specified multiple of the fit standard error or outside a specified prediction intervals. Points can also be selected manually. The data can now be refitted within the Review. All equations are refit and all open windows are updated. Marked points are toggled on and off with successive refits, enabling the impact of excluding points to be more readily appraised.

13. Data Table Reset



In prior versions, once the data table was modified, the original data could only be restored from file. There is now a [Reset XY Data](#) button in the main toolbar and in the [Data menu](#). This makes it possible to denoise or filter a data set and recover the raw data without reimporting it.

14. Improved Residuals



The [Residuals](#) procedure has been enhanced. It is now offered for most TC2D procedures rather than only in the Curve-Fit Review. A numeric window can now be displayed and is automatically updated. The histogram can now be exported.

15. New Connect Options in Graphs



All graphs containing discrete data now offer the option of [connecting the data points](#) with either a cubic spline or a cubic NURBS, in addition to a simple line. By default, imported data are now rendered with connected lines.

16. Differencing, Cumulatives, Normalizations



A [Difference](#), [Cumulative](#), [Normalize](#) procedure has been added. This puts most of the basic data manipulations operations in one place. Many of the operations are new to v5.

17. Generating Data



Prior to v5, data could be generated only using one-line View Function(X) expressions and by applying one-line calculations to the data table. The [Edit menu](#) now offers a [Generate Data](#) option. Apart from the absence of adjustable parameters, the format is similar to UDFs. You can thus generate a data set from complicated expressions

18. Importing Equation References as Data Sources

It is now possible to import references saved from TC2D v5 as a data set. You must specify starting and ending values as well as either an increment or data count. Data are then generated and imported.

19. More UDF Fn#s

For complicated UDFs, additional Fn#s can now be used. The allowed F1 through F9 function assignment in UDFs has been extended to F1 through F24.

20. XY Sectioning



For sectioning, there is a new XY sectioning mode. Here the mouse encloses a region in XY space that will either be included or excluded in the sectioning.

21. Preferences

TC2D v4 offered separate Review Preferences and General Preferences options. TC2D v5 combines these into a single Preferences option available from either the Edit menu or the File menu of the Curve-Fit graph.

22. Additional Mouse Functionality

If the Curve-Fit graph is the active window, you can use the rotating wheel on a wheel mouse to control movement through the equation list. Rotate the wheel forward (away from you) to advance up the equation list and back (toward you) to step down the equation list. You can also single click on the middle button of a three-button mouse (or depress the wheel of a wheel mouse) to advance or go back one equation, depending on the current direction through the list. The equation list window can be toggled on and off by double clicking the middle button or wheel.

23. Automatic Window Placement in Review



The Curve-Fit graph's control panel (or Window menu) now offers six predefined tile options to automatically open and place various Review windows.

The Window menu also offers options to save and import the window state (the windows currently open and their positions).

24. Revisions to Toolbars and Menus

The main toolbar and process toolbar can now be toggled on and off separately. This toggle has been moved to the Edit menu. The main toolbar now contains additional smoothing, denoising, and filtering options. The estimation options have been moved to the process toolbar. Some of the button images associated with denoising, smoothing, filtering, and estimation have been revised.

The Calculate, Table, View, and Non-Parametric menus have been replaced with Data, Filter, and Estimate menus. The items that modify the data table, but do not smooth, denoise, or filter the data are in the Data menu. The procedures that smooth, denoise, and filter the data are in the Filter menu. The items that offer interpolation and prediction are in the Estimate menu.

25. On-Line Documentation

Previous versions of TC2D offered both on-line help and a printed user's manual. Due mainly to memory considerations, the on-line help had been less detailed and lacked typeset equations, descriptions and graphs of the individual non-linear equations, quick-start information, tutorials, and other reference information.

With v5, all information is fully electronic. The on-line help system has been vastly expanded and now includes all reference information previously available only in the printed documentation. As a plus, the on-line help system exclusively uses color graphs to better illustrate the influence of parameters, intervals, residuals, and other aspects of fitting.

The tutorials are also electronic and have been increased in number. Several cover the new or expanded features in v5. These can be accessed directly from the [Help menu](#).

Also included in the v5 distribution is the Adobe Acrobat reader and several key PDF documents. These include a document that lists all linear equations, another that details each non-linear model, and the curve-fitting pitfalls document. These have been reformatted in a standard letter size and use color graphs.

26. Key Features and Enhancements Added to v4

If you are upgrading to v5 from v3, the following are the key items associated with the v4 release:

- 88 new polynomial equations and 110 new rational equations:
 - [High Precision Polynomials and Rationals](#)
 - [Chebyshev Polynomials and Rationals](#)
 - [Fourier Series Polynomials and Rationals](#)
 - [Constrained \(No-Singularity\) Non-Linearly Fit SVD Rationals](#)
 - [Inverse Polynomials](#): 6th through 10th order, Equations 6504-6508
 - [Inverse Log-X Polynomials](#): 6th through 10th order, Equations 6605-6609
 - [Half-Order Polynomials](#): 2.5 through 5.0 max power, Equations 6701-6706
 - [Even-Order Polynomials](#): 4 through 20 max power, Equations 6721-6729
 - [Even-Order Rationals](#): 2/2 through 10/10 max power, Equations 7401-7409
 - [Half-Order Rationals](#): 0.5/0.5 through 2.5/2.5 max power, Equations 7501-7509
- 24 new peak models:
 - [Laplace](#) (Eqn 8168, 8169)
 - [Error](#) (Eqn 8170, 8171)
 - [Student t](#) (Eqn 8172, 8173)
 - [Log-Normal 4-parameter](#) (Eqn 8174, 8175)
 - [Extreme-Value 4-parameter Tailed](#) (Eqn 8176, 8177)
 - [Extreme-Value 4-parameter Fronted](#) (Eqn 8178, 8179)
 - [Inverted Gamma](#) (Eqn 8180, 8181)
 - [Chi-Square](#) (Eqn 8182, 8183)
 - [F-Variance](#) (Eqn 8184, 8185)
 - [Pearson IV](#) (Eqn 8186, 8187)
 - [Exponentially-Modified Gaussian](#) (Eqn 8164, 8165)
 - [Half-Gaussian-Modified Gaussian](#) (Eqn 8166, 8167)
- 2 new transition models:
 - [Cumulative Exponentially-Modified Gaussian](#) (Eqn 8188, 8189)
- Support for Up to 100 External Fortran or C Functions
- [Implicit Model UDF Fitting](#) (v4.02+)
- [Ligand UDF Library](#) (v4.02+)
- [Michaelis-Menten UDF Library](#) (v4.02+)
- New graphics engine:
 - [Common architecture](#) for all program graphs
 - [Y2 axis](#) that can plot residuals, derivatives, or cumulative area
 - [Spreadsheet block copy](#) for all data in a graph
 - [SNP Residuals Graphs](#)
 - [Error Bars](#) with Confidence and Prediction Intervals for [Replicates](#)
- [Graphical XY Column Selection](#)

- Binary Import
- Sort Equation List by equation number
- Visual Basic, Fortran 90, and 80-bit C Code Generation
- Poles of Rational Models in Numeric Summary
- 99.9% and 99.99% Confidence Limits

Uninstalling TableCurve 2D

To uninstall from any Windows NT or Windows 2000 environment, users are required to log in with administrator privileges.

- From the [Start](#) menu choose [Settings](#) and then [Control Panel](#).
- Select [Add/Remove Programs](#).
- Locate [TableCurve 2D v5.0](#) in the list of applications and select it.
- Click the button containing the word [Remove](#).

Quick Start

Automated Fitting

TableCurve 2D has been designed to automatically fit large numbers of candidate curve-fit equations in a fully automated fashion. In general, there are four basic steps:

1. Import or Edit an XY Data Table



The **Import** option in the File menu is used to read a new data table from a disk-based file. **Supported formats** include ASCII, Excel, Lotus, Quattro, SigmaPlot, DIF, SPSS, Systat, and dBase.



The **TableCurve Editor** option in the **Edit menu** is used to enter a data table within TableCurve 2D.

2. Process the Data for the Best Equations

The following **Process menu** fitting options utilize pre-defined equation sets:



Curve-Fit All Equations



Curve-Fit Linear Equations



Curve-Fit Simple Equations



Curve-Fit Robust Straight Line



Curve-Fit Poly/Ratl Equations



Curve-Fit Peak Functions



Curve-Fit Transition Functions



Curve-Fit Kinetic Equations



Curve-Fit Waveform Equations

3. Select the Optimum Equation for the Application

TableCurve 2D offers a large number of **graphical** and **numerical** tools to enable you to choose the **best equation** for your particular needs.



For selecting the most appropriate equation, the Review menu's **Graph Start** option emphasizes the **Curve-Fit Graph**.



The Review menu's **List Start** option emphasizes the **Equation List**.

4. Generate the Printed, File, or Code Output



The **Excel**, **SigmaPlot**, **Lotus 123**, and **ASCII** items in the **Review** are the most commonly used file export options.



The **Print** button in the **Curve-Fit** or **Residuals Graph** toolbar is used to generate printed graphs.



The **Code Generation** option is used to generate the fitted equation in C, Fortran, Pascal, or Basic.



The **Numeric Summary** offers a full statistical report.



The **Residuals** option offers five different formats for viewing and exporting the residuals from a fit.

Questions and Answers

Data Table I/O

- How do I import data from external files?
- How do I paste in the data table from the clipboard?
- How do I append data to the existing data table?
- What is the best way to manage replicates?
- How do I import data from fixed length data block binary files?
- How do I implement an averaging digital filter upon import?
- How do I drag and drop files into TableCurve 2D for import?
- How do I save the data table to a spreadsheet format?
- How do I copy the data table to the clipboard?

Data Input

- How do I input data while simultaneously graphing it?
- How do I reverse the X and Y values?
- Although TableCurve 2D sorts the data by ascending X, is there a way to save the data unsorted or in some other sort order?
- My X values are equally spaced. Do I need to enter each of these?
- Can I have a transformation applied after entering X or Y values?
- Can I enter or edit the data table using a simple ASCII editor?
- What is the best way to assign weights to data?
- How do I generate data?

Viewing Data

- How do I view a graph of the current data table?
- Can I view arbitrary functions of X?
- How do I compare different data sets?
- Can I get a printout of the basic data table statistics?

Data Preparation

- How do I disable points so that they are excluded from the fitting?
- How do I transform one or more of the data columns?
- How do I clear the current data table?
- Can I discard inactive points?
- How do I change the basic data table titles?
- How do I difference the data, create cumulatives, or otherwise normalize the data?
- Can I subtract a baseline?

Filtering Data

- What is the easiest way to smooth or denoise my data?
- Why are there other procedures for smoothing and denoising?
- How do I isolate data components in the frequency domain?
- Apart from parametric fitting and Fourier filtering, can data components be isolated in any other way?

Estimation and Prediction

- When should I use non-parametric estimation?
- Why are there five different five non-parametric estimation options?
- How do I predict or forecast data?
- Can I forecast in both directions?
- How do I optimizing the AR Fit?
- Can I apply the AR Filter to other data sets?

Curve-Fitting

- How can I viewing the built-in equations?
- How do I decide which equations to fit?
- Could you furnish an overview of the built-in fitting options?
- What is a 'high-precision' polynomial or rational equation?
- What is a 'Chebyshev converted' polynomial or rational equation?
- How do I set the curve-fitting preferences?
- Why are there different matrix methods for linear fitting?
- When should I use robust fitting?
- How do I create a custom equation set?
- As soon as I load a data set, some kind of fitting seems to be taking place. What is going on?
- Is this the same thing as the 'Fit in Background' option that appears in the curve-fit progress dialog?

User Defined Equations

- How do I add my own equations to TableCurve 2D?
- If I want to create a User-Defined Function, where do I start?
- In the UDF Adjust procedure, what do I do when the UDF curve is not visible?
- Can I have multiple collections of UDFs?
- Can I express the minimum and maximum limits for parameters with formulas rather than values?
- Is there any advantage to using the built-in non-linear functions within UDFs?
- How do I fit only the UDFs?

Review and Equation Selection

- What is the Graph Start and List Start all about?
- How does one effectively inspect equations?
- I see three tenth order polynomials in the equation list, all with the same goodness of fit. Which one should I use?
- Why is the format for the Lotus 123 export different from that if the Excel export?
- Why is the formula for certain equations missing from Excel and Lotus export?
- How do I display the Curve-Fit and Residuals in the same graph?
- What is an SNP plot anyway?
- Can I set up a reference from a separate fit defining a standard, and have this plotted for all future fits?
- I know which TableCurve 2D equation number I want. How do I find it in the equation list?
- How do I limit equations to only those which are continuously increasing or continuously decreasing?
- Why do some of the equation list options appear in both the List menu in the Curve-Fit graph and in the menu of the Equation List window?
- In the code generation option, is the function code itself portable to different compilers?
- How do I change the numeric precision used in graph titles and labels?
- How do I control which of the titles are displayed in Review graphs?
- Can I copy the curve-fit graph to the clipboard not as an image but as the raw numeric data which

comprises the graph?

- On certain fits the first equation does not appear in the list upon entering the Review. What happened to it?
- Can a Singularity Constrained Non-Linear SVD Rational equation (Eqns 7901-7939) have a pole in the middle of the data range? Also, for equations with no poles in the data range, shouldn't these equations produce the same fit as the linear form?
- If I see points during the Review that I wish to have excluded from a refit, how do I proceed?

Data Table I/O Q&A

How do I import data from external files?



Use the **Import** option in the **File menu**. You may optionally specify a weights column or a column containing the standard deviation of replicate measurements. Use the **Import Preview** to view the data graphically before importing it.

How do I paste in the data table from the clipboard?



Use the **File menu's Import Clipboard** option. The clipboard block can contain any number of columns. Select the desired data table **columns**. You can paste in data from ASCII, Excel, Lotus, Quattro, and SigmaPlot **formats**.

How do I append data to the existing data table?



Use the **File menu's Append** option. Also available are **Append Digital Filter** and **Append Clipboard** options.

What is the best way to manage replicates?



Import a single data set in which all of the replicates have been combined, or **Append** the individual data sets.



Save this data set as the source.

Use the **Process Replicates** option in the **Data menu**. This approach automatically implements weights based upon the individual standard deviations and enables full confidence intervals on the resulting error bars. Fits to the unprocessed source file can also be made for determining the **pure error and lack of fit**.

How do I import data from fixed length data block binary files?

Use the **Import Binary** option in the **File menu**. Then specify the offsets into the file header or repeating fixed length data block in order to read titles and data. Fixed length data block binary files are written by a great many computerized laboratory instruments.

How do I implement an averaging digital filter upon import?

Use the **File menu's Import Digital Filter** option. Start at any location in the data stream and average up to 999 points per table entry. The **Import Digital Filter** option is primarily intended for importing data tables larger than the program's 65536 data point maximum. When the data source is less than this maximum, the preferred approach for reducing the size of the data table is to use one of the **Estimation** options. These offer both data reduction and data enhancement (increasing the data table size). They also can create an arbitrary size output stream.

How do I drag and drop files into TableCurve 2D for import?

Drag the file onto the main TableCurve 2D window or onto the status window. The import procedure is then automatically initiated for this file. If you drop multiple files, the last selected file is read and the remainder are added to the **previous files** list at the bottom of the **File menu**.

How do I save the data table to a spreadsheet format?



In addition to the default ASCII format, the **Save As** option in the **File menu** offers Excel 2000/97, Excel 95, WK3, and WK1 data formats.

The most space efficient format is the SPSS SAV format. This format uses an internal compression, preserves full binary precision, and also saves weights and the status of individual points.

How do I copy the data table to the clipboard?

Use the **Edit menu's Copy** option. The clipboard will contain both ASCII and spreadsheet data formats.

Data Input Q&A

How do I input data while simultaneously graphing it?

 Use the **TableCurve Editor** option in the **Edit menu** and activate the **Graph** button after 3 or more points have been entered.

How do I reverse the X and Y values?

 This can be done in the **TableCurve Editor** using the **Reverse XY** button.

Although TableCurve 2D sorts the data by ascending X, is there a way to save the data unsorted or in some other sort order?

 TableCurve 2D automatically sorts the data by ascending X when exiting the **TableCurve Editor**. To save the data as it is entered, or for a different sort order, the data must be saved from within the TableCurve Editor. The editor contains a **Sort Table** option which can be based on either X or Y, and which can be either ascending or descending.

My X values are equally spaced. Do I need to enter each of these?

 Turn on the **AutoEntry X** in the **TableCurve Editor**. After two X values have been entered, all subsequent entries will be automatic.

Can I have a transformation applied after entering X or Y values?

 Use the **Calculation** option in the **TableCurve Editor** to enter the transform. Be sure **Apply New** is checked.

Can I enter or edit the data table using a simple ASCII editor?

Use the **ASCII Editor** option in the **Edit menu**.

What is the best way to assign weights to data?

Weighting of fits is used to compensate for a variety of factors. The most common is to weight the data based upon uncertainty. When each point represents the average from multiple observations, weights are generally set to inverse variances ($1/SD^2$). When each point represents a single observation, the weights can represent a measure of experimental uncertainty for that element of the data.

Weighting is sometimes used to compensate for limitations within the least-squares process, especially when the Y values span several orders of magnitude or more. It is often simplest and most effective, however, to apply a $Y=LN(Y)$ calculation prior to fitting rather than using weighting.

How do I generate data?

 Use the **Generate Data** option in the **Edit menu** to automatically generate a data set from a set of mathematical expressions. Gaussian noise can be automatically added.

Viewing Data Q&A

Viewing Data

How do I view a graph of the current data table?



The **View Data** option in the Data menu offers a simple graph of the data as well as a very fast numeric list facility.

Can I view arbitrary functions of X?



The **View Function(X)** option is used to plot up to five different functions of X.

How do I compare different data sets?



The **Compare with Reference** option graphs the current data set with up to four reference data sets. A reference can come from any of the supported **import formats**, including **equation references** that have been saved to disk.

Can I get a printout of the basic data table statistics?

Use the **XY Statistics** option in the **Data menu**. A full range of print and copy options are available.

Data Preparation Q&A

How do I disable points so that they are excluded from the fitting?



The **Section Data** option in the **Data menu** isolates portions of a data stream for analysis. You can isolate sections graphically by XY and X sectioning modes or you can also numerically specify ranges to include or exclude. Toggling a point on and off is as simple as clicking on that point with the mouse.

How do I transform one or more of the data columns?



Use the **Data menu's Enter Calculation** option. The X column is treated as a vector with the name X, the Y column with the name Y, and the weight column with the name W. To convert X values in cm to micrometers, for example, enter $X*1E4$ in the X= field.

All of TableCurve 2D's built-in functions are available in a simple **function insert help** where the desired functions are automatically inserted into the expression.

How do I clear the current data table?



Use any of the **Import** options in the **File menu**, or the **Clear All Data** option in the **Data menu**.

Can I discard inactive points?

Use the **Clear Inactive Points** option in the **Data menu** to delete all inactive points from the current data table.

How do I change the basic data table titles?



Use the **Modify Titles** option in the **Data menu**.

How do I difference the data, create cumulatives, or otherwise normalize the data?



Use the **Difference, Cumulative, Normalize** option in the **Data menu** to difference the data with adjustable order and lag, compute various cumulatives, or normalize for unit area, unit power, unit standard deviation, or zero mean.

Can I subtract a baseline?

The **Subtract Imported Reference** option will allow any reference set, with identical X-values, to be subtracted from the current data. This is generally used to subtract a baseline or background, or for fitting the difference between two different data sets.

Estimation and Prediction Q&A

When should I use non-parametric estimation?

If your data have a very unusual profile, it is possible that none of TableCurve 2D's built-in equations will fully describe the features within the data. In these cases, a **non-parametric fit** may be successful.

If you need only a few simple interpolations, a **non-parametric procedure** offers near immediate processing and the same **evaluation procedure** as that of the parametric fitting.

If data are non-uniformly spaced or if you have been forced to disable one or more points containing artifacts, the data can be restored to an equal X-spacing.

Why are there five different non-parametric estimation options?

Each of the procedures has its particular strength.



The **Spline Estimation** option offers three spline procedures for interpolation and five spline algorithms for smoothing. First and second derivatives are also available. This procedure is useful for creating uniformly spaced data sets since uniform data are not required. An option offering user-defined knots can be used to finely tailor a least-squares spline fit.



The **Local Regression Spline Estimation** option combines an adjustable order Loess-type (locally-weighted least-squares) smoothing algorithm with a B-spline interpolant. This procedure is also useful for creating uniformly spaced data sets since uniform data are not required. Although this procedure is much slower than a smoothing spline, the local weighting may result in a more accurate estimation.



The **Fourier Estimation** option offers interpolation and smoothed estimations based upon the **frequency spectrum**. Although very slow with large data sets, this option may be more useful for oscillatory data.



The **Smoothed Data Spline Estimation** procedure combines automated smoothing and B-spline estimation. This is not a smoothing spline, but is rather the fitting of an interpolating B-spline to data that have been pre-smoothed.



The **Savitzky-Golay Spline Estimation** procedure combines the full capability of the Savitzky-Golay smoothing with a constrained spline interpolant. This procedure has been especially tailored for estimating derivatives.

The **Spline Estimation** option is a very good place to start if only interpolation or modest smoothing is needed. The **Local Regression** procedure offers smoothing and possibly a greater accuracy as a consequence of the locally-weighted fitting. The **Fourier Estimation** is useful for estimating data with oscillatory components. It also offers Fourier filtering and estimation in a single step. When significant smoothing is needed, the **Smoothed Data Spline Estimation** is recommended. When an accurate smoothed derivative is needed, you will probably want to use the **Savitzky-Golay Spline Estimation** procedure.

How do I predict or forecast data?



The **AR Modeling and Prediction** procedure offers effective autoregressive forecasting and extrapolation. The AR algorithms include SVD (singular value decomposition) procedures for in-situ noise

removal. This option can be used directly for forecasts or to generate predicted data that are then fit to parametric models.

Can I forecast in both directions?



The **AR Modeling and Prediction** procedure offers three algorithms for predicting ahead, three for predicting earlier values in time, and three that predict in both forward and backward directions.

How do I optimizing the AR Fit?



For SVD-based algorithms, **Signal-Noise thresholding** is graphical and often straightforward. Once the signal space has been properly specified, model order becomes less important. The higher the model order fitted, however, the less the noise will influence the AR fit.



For non-SVD algorithms, multiple orders can be simultaneously plotted to assist in optimum model order selection. **Model order selection criteria** are also available as is an option to **plot complex roots**. Stabilizations are available for roots that lie outside the unit circle.

Can I apply the AR Filter to other data sets?



An AR coefficient set (filter) can be saved to disk and imported and applied to subsequent data sets. You can import as many saved filter sets as you choose and all will be added to the algorithm list using the file name.



The AR filter used for the prediction can also be saved to language code.

Filtering Data Q&A

What is the easiest way to smooth or denoise my data?



The **Automated Smoothing** option in the **Filter menu** is the easiest way to remove noise from the data. An **AI Expert** estimates the optimum smoothing level for each of the six automated procedures. Uniform data are automatically generated for procedures requiring constant X-spacing.

Why are there other procedures for smoothing and denoising?

For optimizing smoothing and denoising, three of the algorithms can be finely tuned in separate procedures. These require a constant X spacing:



The **Savitzky-Golay Smoothing** procedure offers effective time-domain smoothing for data sets with uniform X-spacing. The algorithm offers adjustable order, automatic sequential passes, and optional first through eighth smoothed derivatives. This is the recommended procedure for generating a smooth derivative.



The **Fourier Denoising** option is a specialized Fourier filtration procedure that sets either a frequency threshold for low pass frequency domain filtration, or a signal threshold for zeroing all spectral elements below a given power. The time domain data are reconstructed using the inverse FFT.



The **Eigendecomposition Denoising** option accomplishes a similar function except that the filtration occurs by zeroing those eigenmodes that contain noise. By using a high order decomposition, it is often possible to remove nearly all of the noise within a signal. This is often the best procedure for removing noise from a data set.

How do I isolate data components in the frequency domain?



You can manually edit the **frequency representation** of the data using the **Fourier Filtering** option in the **Filter menu**. This option offers real-time Fourier domain filtering and component isolation. This procedure supports **exact-N FFTs** and **data tapering windows** so that low power components can be isolated and reconstructed.

Apart from parametric fitting and Fourier filtering, can data components be isolated in any other way?



The **Eigendecomposition Filtering** option offers full eigenmode filtering and reconstruction. **Eigendecomposition** is a non-parametric procedure that partitions by signal strength rather than by frequency. In addition to the data, the reconstruction can optionally consist of the eigenvectors, the principal components, the data components, FFTs of the data components, or an FFT spectrum of the data.

Curve-Fitting Q&A

How can I view the built-in equations?

The **Help menu** has a **List Equations** option that lists every built-in equation within the product. These are numbered and grouped by type.

There are help items associated with the various **linear** equation types and for each **non-linear** equation.

The product ships with Adobe Acrobat PDF files. Included is **lineareq.pdf**, a document listing all linear equations, and **nonlineq.pdf**, a document detailing each non-linear equation.



The **Edit Custom Equation Set** option offers the means to customize an equation set from the full set of built-in equations.

How do I decide which equations to fit?



All Equations: TableCurve 2D assumes that it is not often easy to know which equations may be appropriate to a given data set. In the beginning, until you gain certain experience with the product, it is a simple matter to fit **All Equations** to the data.

All of the other fitting options for built-in equations represent conveniences to save fitting time. Since the **Review's equation list** can be filtered and sorted in a number of ways after the fact, it is not necessary to limit the list of candidate equations by restricting the set being fitted.

Where can I find an overview of the built-in fitting options?



All Equations: fits all 3667 built-in equations. This is the recommended option until you become more familiar with the equations.



Linear Equations: all 3491 linear equations; omits the 176 built-in non-linear equations. This is often a good option when searching for an approximating function.



Simple Equations: fits all 2-parameter linear equations as well as non-linear power and exponential functions.



Robust Straight Line: fits a simple straight line equation by least-squares and three robust non-linear procedures. This is generally used when outliers are present or suspected and only a straight line fit is desired.



Poly/Ratl Equations: All polynomials and rationals, including high-precision, Chebyshev, and Fourier series.



Peak Functions: All built-in non-linear peak functions (up to 74 depending on intercept options).



Transition Functions: All built-in non-linear transition functions (up to 29 depending on intercept options).



Kinetic Equations: All built-in non-linear kinetic models (up to 58 depending on intercept options).



Waveform Functions: sine and sine-squared equations.

What is a 'high-precision' polynomial or rational equation?

For Intel coprocessor precision, it is wise to limit polynomials to order 10 and rationals to orders 5/5. TableCurve 2D offers a special class of polynomials through 20th order, and rationals through 10/10th order. In order to produce effective fits of these higher order models, TableCurve 2D performs all fit computations as well as the confidence/prediction intervals evaluation using an entirely separate 38-digit high-precision 32-bit assembly language math emulator. Function evaluation is with the Intel FPU.

If you are fitting full precision generated data, these equations have the potential for producing fractional residuals near the 1E-18 limit. While the high-precision software floating point is exceptionally fast, it is still a matter of software emulation, therefore, the fitting of these equations may appear to be considerably slower than those fitted using the Intel floating point unit. This may be particularly evident when confidence or prediction intervals are being graphed.

What is a 'Chebyshev converted' polynomial or rational equation?

A Chebyshev series polynomial can be reduced to a standard polynomial, one that can be evaluated in about half the time. TableCurve 2D includes this option for those instances where the precision losses certain to be a part of this conversion can be tolerated. The Precision Summary is used to ascertain the extent of these losses.

How do I set the curve-fitting preferences?



The **Curve-Fit Preferences** option in the **Process menu** is used for setting the preferences for all fitting options except the **Peak Functions**, **Transition Functions**, and **Kinetic Equations** which have their own separate configuration.

Why are there different matrix methods for linear fitting?

All linear fits produce an exact least-squares solution only with unlimited precision. In the finite precision world of digital computing, each procedure has its own strengths and weaknesses. In general, there are two major types of algorithms: SVD and the more conventional procedures. Unless you have specific experience with matrix methods, the default **Fast Std** is recommended.

When should I use robust fitting?

There are three **robust procedures** available for non-linear fitting. **Robust fitting** is of value when there is a large dynamic range on the Y-variable (spanning several orders of magnitude or more), and the lowest Y value points are not being sufficiently fitted. It is also of use in managing outliers which adversely affect the fit. The Lorentzian minimization is highly effective and the fastest converging of the robust algorithms. Unless one of these factors is present, least-squares should generally be used.

How do I create a custom equation set?



For a set containing only the equations you want to have fitted, use the **Process menu's Edit Custom Equation Set** option to configure the desired equations.



You can use the **Curve-Fit Custom Equation Set** option to fit this set at anytime.

As soon as I load a data set, some kind of fitting seems to be taking place. What is going

on?

On a multithreading operating system, it is possible for TableCurve 2D to perform two quite independent operations at the same time.



The **Background Thread Fitting** option is used to set up a certain intelligence whereby TableCurve 2D can begin fitting the data the moment it has been imported. This can be used to eliminate a good deal of waiting if you generally fit a specific equation set option.



The light bulb icon is lit when background thread fitting is underway.

Is this the same thing as the 'Fit in Background' option that appears in the curve-fit progress dialog?

No. The **Fit in Background** button merely minimizes TableCurve 2D while a fit is in progress so that you can work in another application.

User Defined Equations Q&A

How do I add my own equations to TableCurve 2D?

 There are two options. The main approach consists of **user-defined functions** which use a simple mathematical syntax and are internally compiled for rapid fitting. Up to 15 user defined functions can be active and fit at any given time. Graphical adjustments allow starting estimates to be highly refined to insure convergence in fitting.

The other option is to create a 32-bit DLL using the C or Fortran language. These are known as **TCEX.DLL equations**, and you may add up to 100 equations in this fashion. The disadvantages of this approach include the required programming skill and the lack of any graphical means for refining starting estimates. The advantages include the exceptionally fast execution speed such compilations offer, as well as the means to create equations and constraints of unlimited complexity.

If I want to create a User-Defined Function, where do I start?

Use the **Process menu's User Functions** option to open the UDF dialog. All 15 of the active UDFs are controlled from this one central point. You may wish to load one of the sample UDF libraries (a set of up to 15 UDFs), or a single UDF sample file into the current UDF position. In order to enable an effective compilation of user functions, some simple rules define how a UDF must be expressed. Many of these are evident from the simple inspection of an example UDF. More information is available in the help system and printed documentation.

In the UDF Adjust procedure, what do I do when the UDF curve is not visible?

 When graphically adjusting UDFs, your starting estimates may be so far off that the UDF curve doesn't even show up with the data. In this case, open the **Scaling** option for the graph and change the Y priority to **Fn**. This should allow you to at see the actual UDF, provided it is defined within the X range of the data. Once you have adjusted the parameters to bring the UDF into the Y range of the data, the **Find and Update** option can often refine the estimates automatically.

Can I have multiple collections of UDFs?

Save each set of UDFs as a UDF library and read in the desired UDF library within the **User Functions** option prior to fitting.

Can I express the minimum and maximum limits for parameters with formulas rather than values?

Parameter limits can be specified with formulas rather than values. For example, YMIN and YMAX can be used as the two limits for a parameter.

Is there any advantage to using the built-in non-linear functions within UDFs?

These **built-in functions** will run appreciably faster than the equivalent code within a UDF. These functions also contain conditional code to insure their definition at all values of X.

How do I fit only the UDFs?

 Use the **Curve-Fit User Functions** option in the **Process menu**. The fitting can also be initiated from the main **User Functions** dialog.

Review and Equation Selection Q&A

What is the Graph Start and List Start all about?

The **Review** utilizes two separate 'desktops', the specific organization of windows used to review and select the most appropriate curve-fit equation.



The **Graph Start** option assumes you are mainly interested in making the selection by stepping through the curve-fit graphs. The default equation list is in a small-popup window that coexists on screen with the graph.



The **List Start** option assumes you wish to focus considerable attention upon the inspection of the equations within the list. The default equation list is large, and automatically closes after a selection is made. Clicking on the List button restores this large list.

Each option takes shape as you customize it to suit the amount of screen space available and the items you wish to see displayed when stepping through the equations.

How does one effectively inspect equations?

Although equation inspection and selection is still somewhat of an art, some guidelines can be given:

Curve-Fit Graph

Graphically inspect the **quality of the fit** across the whole data range (and possibly beyond in the case of extrapolations). The numerical **goodness of fit criteria** are sometimes poor indicators of fit for points at the lower portion of the Y-scale. The zoom processes and logarithmic axes often aid in this inspection.

Detecting Outliers

The coloring of points based upon residuals will readily illustrate points outside 2 and 3 standard errors. These may represent outliers which should be excluded from fitting or treated by a robust non-linear procedure.

Math Errors

Be very cautious of selecting an equation where a part of the data region shown is undefined or where major changes in the curve occur. When a math error or singularity is detected, a vertical line will be drawn at this X-value. A vertical error line is not always drawn when a singularity is present. A curve rapidly going off-scale and then returning, without this error bar, may still indicate a value or region of values where the equation is undefined. In the instance of an extremely sharp singularity, the graph may not show even a hint of the singularity.

Rational Singularities

Singularities are far and away of greatest concern with rational functions. For these functions, refer to the numeric summary to see if any poles (roots of the denominator) exist within the data range or beyond (in the case of extrapolations). When the rational's denominator is a polynomial in x , all real roots are found and reported. When the rational denominator is not a polynomial in x , such as is the case for $\ln(x)$, half order, even order, Chebyshev, and Fourier series rationals, a partitioning is used to isolate pertinent real roots.

Confidence and Prediction Intervals

Graphically inspect the **prediction intervals** for the fit. This is a superb way to detect outliers in the data. A good equation should have tight and stable intervals, especially in the area of concern. Be cautious of intervals which generate math errors (vertical lines of a different color). The computation of the intervals is more sensitive to breakdown due to machine floating point precision limitations than the curve-fit equations themselves.

Replicate Error Bars

If you are fitting replicate data, use **error bars** to detect observations that should be excluded.

Residuals Inspection

Graphically inspect the **residuals** across the whole data range. An ideal, especially in parametric models, is that the residuals not only be very small but also random in sign (no systematic trend). Note the runs count in the Y-title, the number of sign changes in the residuals.

SNP Plot

If you plan on using the confidence limits on the parameters, or the confidence or prediction intervals about the fitted curve, inspect the **Residuals SNP plot** to insure that the residuals are normally distributed.

Fitting Speed

Choose a fast equation if you are going to use it in a program where the speed of the code is important. One of TableCurve 2D's unique features is the ability to choose an equation based on the floating point execution speed of its own generated code for that equation. The FP values are shown in the equation list.

Evaluation

Preview the performance of the equation using the **evaluation procedure** at pertinent X-values before arriving at a final selection.

I see three tenth order polynomials in the equation list, all with the same goodness of fit. Which one should I use?

You are seeing a standard polynomial fitted by the Intel FPU, a high-precision polynomial fitted by the 38-digit math emulator, and a Chebyshev polynomial that has been converted to a standard polynomial. Even when all three show an equal goodness of fit within the equation list, there may be significant differences in the **Precision Summary** in terms of errors arising due to coefficient precision truncation. This issue is normally only of concern when fitting full precision generated data.

In general, the high-precision fit will have the most accurate coefficients.

Since no inverse exists for the Chebyshev converted polynomials, confidence and prediction intervals will not be available.

Why is the format for the Lotus 123 export different from that if the Excel export?

Graphs are included in the Lotus 123 worksheet format. Graphs are not generated in the Excel format. Since each of the products import both formats, use the Lotus 123 format when you want graphs exported, and the Excel format when you do not.

Why is the formula for certain equations missing from Excel and Lotus export?

Some of the functions require a full evaluation routine and cannot be coded into a single cell formula. If you use Excel, there is a very attractive alternative, however. For later Excel versions supporting VBA (Visual Basic for Applications), you may wish to add a Basic module containing the curve-fit equation to the spreadsheet. This approach makes the curve-fit function global to Excel just as if it were a built-in Excel function. The on-line help and printed documentation offer step-by-step instructions.

How do I display the Curve-Fit and Residuals in the same graph?



The **Toggle Residuals on Y2 Axis** button offers a fast way to toggle the **Y2 Axis Residuals** on and off.



You can also use the **Set Y2 Axis Content** option in the control panel of the Review's **Curve-Fit Graph**. You can also display derivatives or cumulative area in the Y2 axis of the graph.

Since the Y and Y2 axis share common X values, the Residuals distribution and SNP plots will not be available. For these graphs, you must use the separate **Residuals** option.

What is an SNP plot anyway?

It is the recommended approach in TableCurve 2D for assuring that errors are normally distributed. A **delta stabilized normal probability (SNP)** plot uses an arctangent transformation on both X and Y to produce a normal probability plot that uses a linear scale for both the X and Y axes. On such a plot, perfectly normal errors plot as a horizontal $y=0$ line. Critical limits also appear as horizontal lines.

TableCurve 2D plots 90, 95, 99, and 99.9% critical limit lines on the SNP plot. A 99% critical limit means that in only 1 out of 100 data sets should even a single point violate this limit. You may find the 99% critical limit the most useful. If even a single data point in the SNP violates this 99% limit, it is reasonable to assume that the errors fail this normality test. You should inspect the SNP before attempting to use the parameter confidence statistics or confidence or prediction intervals in any way.

Can I set up a reference from a separate fit defining a standard, and have this plotted for all future fits?

To save the standard, simply use the **Save Current Equation as Reference** option in the Curve-Fit graph's **Reference** menu.



In the subsequent fits, use the **Equation References** option and select the **Add One or More Imported Fits to References** option. The reference will consist only of the fitted curve.

I know which TableCurve 2D equation number I want. How do I find it in the equation list?



The **Search for Specific Equation** button is used to directly select the TableCurve 2D equation desired.

How do I limit equations to only those which are continuously increasing or continuously decreasing?

Filter the equation list by **Continuous Trend First Derivative**. For the kind of profiles unique to transition data, filter by **Constant Sign in First Derivative**. Derivative filtering is configured in the **Set X-Range for Derivative Filtering** option.

Why do some of the equation list options appear in both the List menu in the Curve-Fit graph and in the menu of the Equation List window?

This is a convenience for the instance where the [Equation List](#) window is not visible.

In the code generation option, is the function code itself portable to different compilers?

The [C](#), [Fortran 77](#), and [Fortran 90](#) function code should compile with any compiler.

How do I change the numeric precision used in graph titles and labels?

Use the [Preferences](#) option in the File menu of the [Curve-Fit Graph](#) to change the title precision.

The label precision is set directly in the [TableCurve 2D graph](#), specifically via the [2D View](#) option.

How do I control which of the titles are displayed in Review graphs?

This is also set in the TableCurve 2D graph's [2D View](#) option.

Can I copy the curve-fit graph to the clipboard not as an image but as the raw numeric data which comprises the graph?

Use the [All Numeric Info as Spreadsheet Block to Clipboard](#) option in the graph's [Copy](#) option. This is available for every program graph.

On certain fits the first equation does not appear in the list upon entering the Review. What happened to it?

The [Rationals Fully Defined Xmin-Xmax](#) is initially turned on within the [Equation List](#). This setting filters out equations which were detected as having a singularity in the data range. Such equations are generally undesired. To see the excluded equations, simply uncheck this Rationals Fully Defined Xmin-Xmax item. This can be done in either the List menu of the [Curve-Fit Graph](#) or in the Filter menu in the [Equation List](#).

Can a Singularity Constrained Non-Linear SVD Rational equation (Eqns 7901-7939) have a pole in the middle of the data range? Also, for equations with no poles in the data range, shouldn't these equations produce the same fit as the linear form?

For these [Singularity Constrained Non-Linear SVD Rational](#) models, the fit begins with the coefficients for the equivalent linear fit. If a pole is detected in the X range of the data, the coefficients are adjusted in an effort to produce a model without such poles. Then, a true non-linear rational fit is made, with a constraint set to eliminate, if possible, the formation of any pole within the X data range. In a non-linear fit, if the zeroing of singular values produces nonsense, that iteration's adjustments are ignored. As such, it is generally possible to keep the coefficients within a useful range for the SVD algorithm. This results in the most stable set of coefficients possible.

In general, because of the constraints, the SVD, and the non-linear nature of the rational model, the results from these fits will differ from that of the linearly fitted rationals of the same form, even when no data range singularities are present.

As in all non-linear fitting, constraint-based fitting is hardly foolproof, and this is all the more true when the coefficients must be resolved through SVD. As such, these models are not always successful. In some cases, it is simply not possible to avoid the formation of a pole in the X range for even a modest goodness of fit. When these equations are successful, however, the fit will generally be superior to the linear procedures.

If I see points during the Review that I wish to have excluded from a refit, how do I

proceed?



The **Mark Excluded Points for Refit** option is used to automatically select outliers or points where math errors occur. The outliers can also be tagged manually by left clicking on the point desired. Marked points are redrawn with a surrounding circle. To unmark a point that has been marked, click once again with the left mouse button. The surrounding circle will be removed.

When a refit is initiated, marked active points are made inactive while marked inactive points are made active. You can go back and forth between including and excluding a set of marked points by successive refits.

File Menu



The **File** Menu options offer the means to import a data table into the program from a file, to save the data table to an ASCII file, to view any ASCII file, to reset the program's startup defaults, or to exit the program.

Data Table Import



The **Import** option opens a file and imports an XY data table.

The **Import Digital Filter** option opens a file and imports an XY data table by reading every nth data value.



The **Import Clipboard** option processes the information currently in the clipboard for an XY data table.

The **Import Binary** option processes a fixed block binary data file for an XY data table.



The **Append** option opens a file and appends an imported XY data stream to the current data table.

The **Append Digital Filter** option opens a file and appends an imported XY stream to the current data table by reading every nth data value.

The **Append Clipboard** option processes the information currently in the clipboard and appends an imported XY data stream to the current data table.

Data Table Export



The **Save As** option saves the current data table to disk.

List ASCII File

The **ASCII List** option offers the ability to list, edit, print, and save back ASCII files.

Restoring Defaults

The **Reset All Defaults** option restores all defaults to the installation state.

Other File Menu Options

The **Exit** option closes the program.

A **Previous Files** list will be appended to the bottom of the File menu.

It is also possible to **Drag and Drop Files** into TableCurve 2D.

Import

The **Import** option in the File menu is used to read a new data table into the program from a disk-based file. The supported **import file formats** include ASCII, Excel, Lotus, Quattro, SigmaPlot, SPSS, Systat, TableCurve 2D Fit Reference, DIF, and dBase. This option will read each data point directly into the data table.

Other Import Options

- To average every n points in a data stream starting at a specified index, use the **Import Digital Filter** option.
- To import or append data from the clipboard, use the **Import Clipboard** option.
- To import data from fixed length data block binary files, use the **Import Binary** option.
- To combine the new set with the existing data, use the **Append**, **Append Digital Filter**, or **Append Clipboard** options.

X and Y Column Selection

A **column selection** procedure is offered for the multi-column ASCII, Excel, Lotus, Quattro Pro, SigmaPlot, dBase, SPSS, Systat, and multi-column DIF **import formats**.

Basic Titles

In all import procedures, you will be offered a main, X, and Y title. You may accept or edit these **basic titles**. If you press **Previous Titles** in the titles entry dialog, the titles present prior to the import, if any, are restored.

Import vs. Open

To conserve memory, TableCurve 2D does not include an Open option. A spreadsheet is opened only long enough to retrieve the necessary data vectors and immediately closed. The only exception to holding a file open is an **automation** session that uses an Excel data source. In this case, memory permitting, the Excel file will be maintained in memory until all of the data sets are processed.

If you need to reaccess the same file, the **Previous Files** list at the bottom of the File menu can be used to immediately reopen the file for importing a different data set.

Processing Similar Data Sets

TableCurve 2D is not designed to import multiple data sets at the same time. When multiple sets of a similar type need to be processed, it is much simpler to use the **automation** facility available in all of the program's principal procedures. The automation makes it possible to sequentially input any number of data sets from an Excel file, or via an acquisition DLL, in an unattended fashion.

The recommended approach is to import and process a representative data set. This could be the first data set in an Excel data source that contains hundreds or thousands of sets. Once all the analysis and processing conditions are optimized for this initial set, the automation can then be invoked to process the remaining data sets without requiring any user input.

Import Formats

The following formats are supported for data import:

- **ASCII** (Single, X-Y, and Multi-column)
- **Excel** (Excel 97, Excel 95, v5, v4, v3)
- **Lotus 123** (WK4, WK3, WK1, WKS)
- **Quattro Pro** (WB2, WB1, WQ1, WKQ)
- **SigmaPlot** (JPG, SPW, SP5, SPG)
- **SPSS** (SAV v7.5 and v8)
- **Systat** (SYS v8)
- **TableCurve 2D Equation Reference** (REF)
- **DIF** (Single, X-Y, and Multi-column)
- **dBase** (DBF III+, IV)

XY ASCII Files

This is the default data format. The default XY data file optionally contains a main title, X title, and Y title on the first three lines followed by successive lines of data. Almost any form of data can be converted so long as the X value precedes the Y value and so long as there is a delimiter between the values such as one or more spaces, a comma, a separate line, etc. **Weights** may follow the X and Y value by having an uppercase **W** preceding the weight value.

An X-Y ASCII data file must meet the following conditions:

- The X and Y-values must be sequential. The X-value must come first, followed by the Y-value and there must be some form of delimiter between the values. The first valid number is read into X1, the second into Y1, the third into X2, the fourth into Y2, and so on. The delimiter can be any whitespace character (space, new line, tab, etc.), a comma, or almost any other non-numeric character.
- The first three lines of the file can optionally contain a description of the data, an X-title, and a Y-title (if these begin with a numeric character, the string must be delimited by double quotes).
- Comment strings can exist in the data area, provided these are on a separate line or at the end of a line or numeric data. If a string begins with a number, it must be delimited by double quotes. Do not start a comment with the asterisk (*) or the W character.
- A concluding asterisk (*), alone or with comment text, marks the X-Y data pair as inactive. Such marked points are not processed in TableCurve 2D, although the points can be reactivated in the **TableCurve Editor**.
- A concluding W, followed by a number, is read as the weight for the data point. The numbers must be between 1E-30 and 1E+30. All weights are 1.0 by default.
- TableCurve 2D considers .PRN, .DAT, and .TXT to be default ASCII file extensions.

Typically, the standard TableCurve 2D X-Y ASCII file will look like:

```
Main description
X-title
Y-title
1.234 2.345
3.456 4.567
5.678 6.789
7.890 8.901
9.012 10.123
```

or:

```
1.234,2.345
3.456,4.567
5.679,6.789
7.890,8.901
9.012,10.123
```

All numeric values must be between 1E-30 and 1E+30. Scientific notation uses the standard format. For example, 1.2345E+12, 1.2345e+12, and .12345E+13 are valid entries. Space should not exist anywhere within the characters.

An ASCII data file can be created in any ASCII editor. The ASCII extracts of databases, spreadsheets, and other programs usually require no modification provided only two columns or fields of data are involved. The main problem with extracted or generated ASCII files will usually be how empty or numeric overflow values are handled. If such an X-value or Y-value is represented by a string, TableCurve 2D will skip over it and the proper X-Y-X-Y... sequence will be lost. If a value is stored as 0.0, which is frequently the case, TableCurve 2D will register this point as an active data pair, with possibly unfavorable consequences on the resulting analyses. Such files must be pre-edited.

Single Column ASCII Files

Frequently, the X-data will begin at a given point and thereafter vary with a constant increment. Many instruments, therefore, tend to generate single vector data files for space reasons. If you cannot save an X-Y file directly, TableCurve 2D can read the single column file. If the fourth line of the data file contains only a single numeric value, you will be asked if you wish to do a Single-Column Read. If you choose this option, you must specify the starting X-value and the X-increment so that the X-values can be automatically generated.

Multi-Column ASCII Files

If the fourth line in an ASCII data file contains three or more numeric entries, you are presented with an option to do a multi-column import. If you choose this option, you will be offered a column selection based upon all entries in the column, regardless of whether these are strings or numeric values. You must then select an X and Y column, and optionally a *Weights* column. Use the column number and the ASCII string information in the initial columns to determine the columns desired.

For TableCurve 2D to read a multi-column ASCII file successfully, it must be in one of two formats, both of which require all row entries for a given sequence of columns to be on a single line which concludes with a carriage return:

- **Comma Delimited Format** - Here all values are separated by commas. You may have any number of spaces within the data, but each column entry, including empty positions, must be separated from other entries by a comma. With this format, ASCII character strings do not have to be delimited with quotes. You will have to delimit a string with quotes, however, if it contains a comma.
- **Space and Quote Delimited Format** - In this case, there are no commas separating the different column entries. Rather, TableCurve 2D relies on a space or spaces between the entries of numeric values, and on double quotes beginning and concluding every character string. With this format, quoted character strings are essential. You must also use a pair of empty quotes to represent empty column positions.

The following two examples illustrate both formats and represent valid multi-column ASCII files:

```
Column 1,Column 2,Column 3,Column 4,Column 5
1.234,2.345,Data Missing,7.890,8.901
3.456,4.567,6.789,9.012,10.123
```

```
5.678,6.789,7.890,11.234,12.345
7.890,8.901,,13.456,14.567
```

```
"Column 1" "Column 2" "Column 3" "Column 4" "Column 5"
1.234 2.345 "Data Missing" 7.890 8.901
3.456 4.567 6.789 9.012 10.123
5.678 6.789 7.890 11.234 12.345
7.890 8.901 "" 13.456 14.567
```

It is always a good idea to carefully inspect the input data from multi-column files whose formats may not be fully compatible. The TableCurve 2D multi-column import procedure should read most multi-column ASCII files generated by commercial software.

If the fourth line of an ASCII file contains more than two floating point values, you will be offered a multi-column read option. A multi-column file must either be comma delimited or freeform with quoted strings.

Excel, Lotus, and Quattro Pro Files

The column identifier and the strings present in the first 100 rows are used to aid in selecting the columns desired. For a given column to be offered for selection, there must be at least one cell entry present for that column within the first 100 rows. Across all sheets or pages, the maximum number of columns available for selection is 4096. Use these text strings and the column designation to select an X and Y column, and optionally a **Weights** column for the data table.

The **column selection list** will show both the sheet and column in an Excel-like nomenclature. For example, the column designated as (3)D is column D in the third sheet. For Excel 95, Excel 97, and Excel 2000, the name assigned to the sheet will also be shown following the worksheet number. The following considerations apply to importing data from spreadsheet files:

- All data to be imported must be in numeric or formula format. Numbers saved as strings will not be read. Strings may appear anywhere within the columns.
- For Lotus, Quattro, and versions of Excel prior to Excel 95, TableCurve 2D will seek to read a main title from the first row of the sheet or page. This title will be available only if the first row contains a single string entry. This title must thus suffice for all sets on the page. It can be in any column. With the exception of the title column, TableCurve 2D will offer the first string in each selected column as the X or Y variable title. In the title column, the second string is offered as the variable title.
- For Excel 95, Excel 97, and Excel 2000, the first row can contain any number of main titles, each applicable to the specific data set. This supports Excel 95/97/2000 as a data source in **TableCurve 2D Automation**. It is recommended that you place each data set title in the Y column of that particular set. If the Single X format is used, this means that all columns with data will have a main title in the first row except column A which will contain the X values. For the XY Pairs option, the titles should be in columns B,D,F,etc. TableCurve 2D will offer the row 2 string in each selected column as the X or Y variable title.
- Valid data entries must be present in the columns in order for that row's entry to be added to the data table. If for a given row, the column specified for one variable has a numeric value and the column for another is empty or is filled with a string, no entry will be made to the table.
- All instances of a zero value in an X or Y column will be included in the data imported. Such points, if invalid, can be toggled off in the **Section** or **TableCurve Editor** option.
- You will not be able to select the same column for more than one variable.

- To insure proper X-Y correspondence, all columns are required to reside on a single page or sheet.

When the read operation is completed, you will be shown the number of points read and you will be offered the option of using the titles detected in the spreadsheet as the main, X, and Y titles for the data set. If you press [Previous Titles](#) in this titles entry form, the titles present prior to the read, if any, are restored.

SigmaPlot Files

TableCurve 2D can build its data table from the numeric values stored in SigmaPlot JNB, SPW, SP5, and SPG graph files. The import process is very similar to that of the spreadsheets. There are a few differences:

- Only the first available worksheet is available for import. This can be an embedded Excel worksheet.
- The SigmaPlot column selection list consists of the column labels assigned within SigmaPlot rather than the strings in the initial columns.
- Only numeric entries are read. Embedded strings and missing values are not imported.

SPSS Files

The SPSS SAV format used in v7.5 through v10 is supported for data import. The process is again nearly identical to spreadsheet import. The differences are:

- You must select columns based on the variable name and variable label assigned within SPSS.
- Valid X-Y pairs, those without system or variable-specific missing values, are added to the data table.

Systat Files

The Systat SYS format used in v8 is supported for data import. The differences relative to spreadsheet import are:

- You must select columns based on the variable name assigned within Systat.
- Valid X-Y pairs, those without missing values, are added to the data table.

TableCurve 2D Equation Reference Files

TableCurve Equation REF files contain all information necessary to evaluate the equation. When an equation reference file is selected for data import, discrete data is generated. You must supply a starting and ending X-value and also either an X-increment or total data count.

- Reference fits cannot be imported from v4 and prior releases of TableCurve 2D.

Data Interchange Format (DIF) Files

If a DIF file contains only one column of numeric data, a single-column read is offered. You must specify the starting X-value and the X-increment so that the X-values can be automatically generated as the Y values are read.

If a DIF file contains two columns of numeric data, the file is automatically read assuming the first column to contain the X-values and the second column the Y-values. If the variables need to be reversed, this can readily be done within the [TableCurve Editor](#).

When a DIF file contains more than two columns of numeric information, a multi-column read is offered. You will be offered a column selection based upon a column ID and any title assigned to this column. Just as with the spreadsheet options, you must select a column for the X-variable, for the Y-variable, and optionally for the [Weights](#).

dBase Files

TableCurve 2D can build its data table from the numeric information in dBase databases. Both dBase III+ and dBase IV type DBF files are supported. The import process is similar to that of a spreadsheet except numeric fields are selected for the X and Y variables. The following rules apply:

- Only numeric fields (type N in III+ and types N and F in IV) are offered for selection.
- If a dBase record is marked for deletion, the data from the selected fields will still be added to the data table, but this X-Y pair will be marked as inactive and will not be included in TableCurve 2D processing. Such points, if invalid, can be toggled off in the [Section](#) or [TableCurve Editor](#) option.
- Zero values in dBase data files are sometimes legitimate and other times simply represent a field that has yet to receive a value. All instances of a zero value in one or both of the fields will be included in the data imported, but these data pairs will be marked as inactive. Such points, if representing meaningful data, can be made active by using the [Section](#) or [TableCurve Editor](#) option.

dBase database files do not use a binary format for storing numeric values. Numeric storage is generally in fixed decimal position ASCII strings. This limits the range and precision of floating point values. The import procedure will preserve whatever precision exists within the original dBase numeric field. Under certain conditions, versions III+ and IV can both store ASCII scientific notation. Type N and F fields containing scientific notation will be correctly read by TableCurve 2D.

When selecting columns from a dBase file, you are selecting database field names for the X and Y, and optionally the [Weight](#), variables.

Column Selection

The multi-column ASCII, Excel, Lotus, Quattro Pro, SigmaPlot, dBase, SPSS, Systat, and multi-column DIF options use a common column selection procedure.

- The first selection is automatically assigned to the X column and the second is automatically assigned to the Y column.
- A weights column can then be specified, or the X and Y columns respecified, by selecting the desired column and then clicking on the [X Values](#), [Y Values](#), or [Weights](#) buttons.
- If the [SD Wts](#) box is checked, each [weight](#) value is assumed to consist of the standard deviation associated with multiple Y measurements at each particular X value. When this box is checked, the imported column must consist of SD values. The data table weights will consist of $1/SD^2$. [Error Bars](#) will then be available in the curve-fit graphs.
- To preview the data to be imported, the [Import Preview](#) button must be checked. The graphical preview is automatic. Graphs are shown once both a valid X and valid Y have been selected. The number of valid points is shown just below the graph. The graph's format is fixed.

Import Digital Filter

The **Import Digital Filter** option will successively average the input values for n points and store these averages in the data table. All of the **supported file formats** can be used with the digital filter options. Use these digital filter option when you wish to reduce a large number of data points to a more manageable size for fitting, or when your data set exceeds the program's maximum of 65536 points.

The **Append Digital Filter** option functions similarly, except that the current data is not cleared within the data table.

As an example, digitally filtering a 10,000 point data file with $n=20$ and beginning with the first data point, would result in a data set containing 500 points, each point representing a twenty point average on the raw data.

The n for the filter can range from 1 to 999 and the starting point can be any within the extent of the data. Note that setting $n=1$ and starting at point 1 is the same as using the standard **Import** option.

Note that a simple averaging digital filter is primarily useful for reduction of very large data sets in an import procedure. For a far more accurate reduction or enhancement of data, or to create uniformly spaced X data, TableCurve 2D provides a variety of estimation procedures in the **Estimate** menu.

Import Clipboard

The **Import Clipboard** option in the File menu is used to import the contents of the clipboard into the TableCurve 2D data table. You may import numeric information from text files, or blocks from Excel, Lotus 123, Quattro Pro, and SigmaPlot files.

Clipboard Import vs. Paste

Note that TableCurve 2D does not offer a Paste option in its Edit menu. This is because the data columns of interest are often non-adjacent in the data source. This Import Clipboard option offers the means to **select from all of the columns** contained in the clipboard. This is identical to the **Import** procedure, except that the clipboard is used instead of a file. In fact, this option first saves the data to **clipbrd.wk1**, **clipbrd.wk3**, or **clipbrd.prn**, depending on the format available in the clipboard. The spreadsheet formats are given priority since these maintain full binary precision. The data are then processed either as a WK1 or WK3 file, or as an ASCII file.

Two or More Columns of a Spreadsheet Block

To import a spreadsheet block from the clipboard, there must be at least two columns present. You must then **select an X and Y column**, and optionally a **weights** column based upon column number and content. The data are imported in accord with the **WKx format**, regardless of the spreadsheet source.

Two Numeric Columns of ASCII Text

In this case, the X and Y data are automatically read. Every valid data pair is placed in the data table. If the variables need to be reversed, this can readily be done within the **TableCurve Editor**. The data must follow TableCurve's **XY ASCII format**.

One Numeric Column of ASCII Text

In this instance, you must enter an initial X-value and an X-increment. The X values are then automatically generated as Y values are read. Every point is placed in the data table. This is identical to the **single column ASCII format** in text files.

Multiple Numeric Columns of ASCII Text

You must **select an X and Y column**, and optionally a **weights** column based upon column number and content. The data must follow TableCurve 2D's **multi-column ASCII format**.

Import Binary

The **Import Binary** option is a procedure that can be used to read many instrument-generated binary data files. This is an advanced feature that should be used only by individuals whose skills include a good familiarity with binary data formats. In general, you will need a published specification for the binary file format you wish to configure.

Fixed Length Data Blocks and Fixed Position Offsets

The binary file must contain fixed length data blocks where the values to be read are a fixed offset from the beginning of the block.

Binary Configuration Files

TableCurve 2D uses binary configuration files to store the specific information needed to process a given type of binary data file. You must first create or read a binary configuration file prior to importing the binary data. These configuration files have BCF extensions.

Binary Configurations Supported

TableCurve 2D's binary read option supports two different formats:

- Both the X and Y values are in the repeating data blocks.
- The starting X-value and/or constant X-increment are stored in the file header. Only the Y values are in the repeating data blocks.

Both formats allow for an optional import of a main, X, and Y data title from the file header.

Title Offsets and Lengths

The offsets and lengths must be specified in decimal (base-10) values and are measured from the beginning of the file. The first byte in the file is at offset 0. Title import is optional. Simply leave a title offset blank if you do not wish to import that title. If the title string is null terminated, you need not specify the title length. In this case, the length field can be left blank or you may enter 0.

X in Header Offsets and Lengths

If the X information is in the file header, you must provide an offset and data type for the initial X value and/or an offset and type for the X increment. If the initial X type is None, an initial X of 0.0 is assumed. If the increment type is None, an increment of 1.0 is assumed.

Supported Data Types

The following data types are supported:

<u>C Data Type</u>	<u>Bytes</u>	<u>Data Range</u>
Signed Char	1	-128 to 127
Unsigned Char	1	0 to 255
Signed Short Integer	2	-32768 to 32767
Unsigned Short Integer	2	0 to 65535
Float	4	$\pm 1.2\text{E}-38$ to $3.4\text{E}+38$
Signed Long Integer	4	-2147483648 to 2147483647
Unsigned Long Integer	4	0 to 4294967295
Double	8	$\pm 2.2\text{E}-308$ to $1.8\text{E}+308$

Long Double 10 $\pm 3.4\text{E-}4932$ to $1.1\text{E+}4932$

The binary read option does not support the Pascal 6-byte float, the non-IEEE BASICA floating point format, or the floating point binary decimal format.

Consistent Data Types

All X and Y values must be type-consistent throughout the data blocks. The X type may, however, be different from that of the Y.

Data Block

You must specify the offset for the first data block and the length of a data block. If you are specifying X as being in the data block, you must specify its position within the data block as well as its data type. The Y data position and type in the data block must always be specified. The X and Y positions are relative to the start of each data block.

Read

The [Read](#) option is used to read a previously saved [BCF] binary configuration file. Binary configurations saved under the DOS version of TableCurve 2D can also be read.

Save

Use the [Save](#) option to save a [BCF] binary configuration file.

Initiating the Import

Click on [OK](#) to initiate the binary import. The file you subsequently select will be processed using this binary configuration.

Appending Data

An append option is not available for binary reads. To append binary files, individually import the files and [save](#) them in a supported format. These files can then be sequentially appended.

Append

The **Append** option in the File menu is used to append additional data sets to the current data set. Any number of sets can be appended in sequence. Other than preserving the current data set, this option is identical to the **Import** procedure.

You may use this option if you want to create a standard based upon fitting any number of similar data files.

Appending Subsequent Portions of a Data Stream

If the X-values are different in the appended set, the combined set's X-values may not have a constant X increment. While not a problem for the curve-fitting, this can be a problem for some of the data conditioning algorithms requiring a constant X-spacing. **Uniform X-spacing** can be restored using several estimation options.

Appending Similar Data Sets to Create An Average

To average replicates, you can use the **Process Replicates** option in the Data menu. Do not invoke this option until you have appended all data sets because TableCurve 2D's data table retains only the variance at each X-value, not the number of samples at that X. You cannot append a data stream to a data table that already consists of averaged values (the averaged set and the appended set would be given equal weight in the averaging).

To save the appended sets prior to averaging, you must use the **Save As** procedure before invoking the **Process Replicates** option.



When the replicates are averaged, the original appended series can be restored with the **Reset XY Data** option.



Error bars are available in the Data menu's **View Data** option if replicates are averaged. If all of the replicates for a given X-value are identical, an SD of 0 is used for the error bar. If only a single value is available, TableCurve 2D uses the conservative approach of setting the SD for that point to the range of the Y-variable.

Append Digital Filter

The [Append Digital Filter](#) option in the File menu is identical to the [Append](#) option except that the appended input data stream is decimated by averaging while it is being imported. The decimation is identical to that which occurs in the [Import Digital Filter](#) procedure.

Append Clipboard

The **Append Clipboard** option in the File menu is similar to the **Append** option except that the data source consists of the current clipboard contents rather than a disk-based file. Other than preserving the current data set, this option is identical to the **Import Clipboard** operation.

Save As

The **Save As** option saves the data table to disk.

ASCII Format

The default format is a 15 digit precision ASCII file with a PRN extension. In a saved ASCII file, the titles are always in double quotes and each X-Y data pair is always on a new line. The flag character '*' will follow the Y value if the X-Y pair has been excluded. If **weighting** is used, the weights will then follow preceded by an uppercase W. Spaces are used to separate the data elements. These ASCII data files can be modified with any ASCII editor, including the **ASCII Editor** available from within the program.

Excel 95 and Excel 97/Excel 2000 Formats

You may optionally save the data table as an Excel 95 or Excel 97/Excel 2000 spreadsheet file. These files will offer 15-16 digit binary floating point precision. The first row of the spreadsheet will contain the main title and the second row will contain the X and Y data titles. The X values will be in the first column, the Y values in the second, and if **weighting** is used, the weights will appear in the third column. The flags for active and inactive points are not preserved when saving an Excel file. Only Excel 97/Excel 2000 supports the 65536 data table limit in TableCurve 2D.

WK1,WK3 Formats

You may optionally save the data table as a Lotus WK1 or WK3 spreadsheet file. The WK1 will have 15-16 digit binary floating point precision and the WK3 will have the full 19 digit floating point precision. The first row of the spreadsheet will contain the main title and the second row will contain the X and Y data titles. The X values will be in the first column, the Y values in the second, and if **weighting** is used, the weights will appear in the third column. The flags for active and inactive points are not preserved when saving a WK1 or WK3 file.

SPSS SAV Format

You may also elect to save the data table as a compressed SPSS SAV file. The SAV will have 15-16 digit binary floating point precision. The X variable will use the name XVAR and its label will be the current X title. Similarly, the Y variable will use the name YVAR and its label will consist of the current Y title. A column named ACTIVE will contain 1 if a point is enabled, and 0 if disabled. A WEIGHT column will contain the weights, or values of 1.0 if **weighting** is not present. An SPSS file saved from TC2D will be automatically read upon import (no column selection) if these four variables comprise the file. In addition to offering direct export to SPSS, the SAV format is the most efficient of the available formats for disk storage.

SYSTAT SYS Format

For export to SYSTAT, the data table can be saved as an SYS file. The file will have 15-16 digit binary floating point precision. The X variable will use up to 12 characters of the current X title. Similarly, the Y variable will use up to 12 characters of the current Y title. A WEIGHT column will contain the weights, if **weighting** is present. Since an SYS file does not preserve full title information, it is recommended only for export to SYSTAT.

ASCII List

The [ASCII List](#) option uses the [TableCurve Text Viewer](#) to view any ASCII file of interest. This option permits the reading of successive files. You can also use this option to append ASCII files. Note that you can append files directly to the data table with the various [Append](#) options. The limit of file length is 65536 lines.

Reset All Defaults

The [Reset All Defaults](#) option will restore all of the program's defaults. This means that all custom window positions, selected fonts, color schemes, point formats, and other graph information will be reset. The program is designed to take shape as you work with it. Use this option only if you are dissatisfied with the layout, colors, and formats you have chosen.

Exit

You can use the [Exit](#) item to exit from the program. If the current data table has not been saved, you will be presented an opportunity to do so before the program closes. If this is the primary instance of the program and there are other [supplementary instances](#) running, all instances of the program will be closed.

Previous Files

The program remembers the last five files read into the program and stores their names at the bottom of the File menu. These are saved across sessions. You may select any one of these files to immediately load the file listed into the program. Use this option to reaccess the columns of a spreadsheet containing data for multiple curve-fits.

Drag and Drop Files

You may drag and drop any file into the program. If you drag a single file and drop it either into the main window or into the status window, this file will be immediately read. If you drag and drop multiple files, select no more than five total files. The last file selected will be immediately read. The remainder will be added to the **Previous Files** list at the bottom of the File menu.

Edit Menu



The **Edit** menu contains options to input or edit data, generate data, copy the data table to the clipboard, modify data titles, set general preferences, and toggle the program's toolbars.

Editing Data



The **TableCurve Editor** option offers a mini-spreadsheet editor for modifying or viewing the main data table.

The **ASCII Editor** option offers a simple ASCII editor for modifying and viewing the data table. Note that on Windows 95 and Windows 98 systems, the ASCII Editor has a 64K limit, or approximately 1500-2000 data points. The TableCurve Editor can edit data tables up to the program's 65536 X-Y limit.

Both editors can be used to **weight** data.

Generating Data



The **Generate Data** option offers a powerful parser and expression compiler for generating user-defined data.

Modifying Basic Titles



The **Modify Titles** option is used to edit the main, x, and y titles for the data table.

Copying Data Table to Clipboard

The **Copy** option copies the current data table to the clipboard in both ASCII and binary formats.

General Preferences

The **Preferences** option is used to set the few preferences that cannot be set from within the various procedures.

Toolbars

The **Main Toolbar** toggles on and off the general toolbar.

The **Process Toolbar** toggles on and off the processing procedures toolbar.

TableCurve Editor

The [TableCurve Editor](#) is a spreadsheet-like procedure for the input or editing of the program's data table. Use this option if you wish the convenience of spreadsheet entry or if you need custom X-Y data table entry features such as automatic X entry, application of a calculation to the entered values, or to view a graph of the data as it is being entered.

Editing or Input

Simply place the cursor in the numeric field of interest and enter or modify the entry. You may enter any numeric value, expression, or equation up to 80 characters in length. A numeric entry can be concluded by pressing Enter, by clicking in any other field, by clicking on the Next button, or by clicking on any of the arrow movement controls.

Moving Between Data Fields

To move about the data table, you may use the arrow controls in conjunction with the table's scroll bar. You may also use the standard keyboard cursor keys.

Active and Inactive Data Points

An active data point is included in the curve-fitting. An inactive data point is excluded from the fit. The [Ex](#) column of the TableCurve editor contains checkboxes for excluding individual points. A point is inactive or excluded when the box is checked. You may also graphically toggle points between active and inactive using the [Graphical Sectioning](#) option.

Copy, Cut, Paste

Use these buttons to copy, cut, or paste a single numeric entry. You will use this most often to copy a numeric value that is repeated rather than having to reenter the value. This option cannot be used to paste multiple values into the data table. To append to the data table from the clipboard, use the [Import Clipboard](#) option from the program's main File menu.

Inserting and Deleting Rows

A data row can be deleted by pressing the [Delete](#) button or by pressing Ctl-Y. You must be certain the cursor is in the row you actually want to delete. To insert a row, place the cursor in the row that will follow the inserted entry. Click on the [Insert](#) button or press the Insert key or Ctl-N. You must provide both an X and a Y value before you can move to any other data row. The program's data is automatically sorted by ascending X upon exit from the editor, and as such, you may simply wish to input inserted data at the end of the table.

AutoEntry

Use these options if you wish to have the editor automatically input a given value. This is useful for constant increment X or Y values. You may have either X or Y autoentry on at any given time. This option will determine the value to enter based upon the two previous values. The position in the editor is automatically advanced. The W autoentry for the Weights is not based upon previous values, but rather upon the default value of 1.0 as the weight for each data point. You will wish to leave W autoentry on unless you will be specifically entering weights for the individual data points.

Entering a Calculation

A [Calculation](#) can be entered inside the editor. The calculation is applied to each numeric value at the

time it is entered. It is best to restrict X and Y calculations used in the editor to functions of the respective variable being entered. In other words, an X calculation should only use X and not Y or W, such as $X=X*2.54$. A Y calculation should only use Y and not X or W, such as $Y=Y/2*PI$. A weight calculation can and probably will use Y or X, such as $W=1/LN(Y)$.

Applying the Calculation to Existing Data

After entering a calculation in the editor, you are offered an option to **Apply the Calculation** to all existing entries in the data table.

Toggling a Calculation

If the **Apply Calc** item is checked, the calculation is applied to all new data as it is entered at the bottom of the data table.

Viewing the Data Being Entered

The **Graph** button is used to open a separate window which displays the data as it is entered in the editor. The graph is automatically rescaled as new data is entered or when old data is deleted. Use this option to insure your data is being entered properly.

Saving the Data Table

This option saves the data table to an ASCII file. This is the program's default data format. Unlike the save option in the File Menu, this option preserves the actual ordering of the data points.

Clearing the Data Table

The **Clear** button is used to clear the data table of all entries.

Basic Titles

Use the **Titles** button to reassign the main, X, and Y titles for the data table.

Sorting the Data Table

The data table can be sorted by ascending X, descending X, ascending Y, or descending Y. You must save the sorted table inside the editor in order to preserve this ordering. The data table is automatically sorted by ascending X values in the prescan which occurs when exiting the editor. All data tables saved from the program's main menu will have this ascending X ordering.

Reversing X and Y Values

If you need an explicit equation for X as a function of Y, you will need to use this option to reverse the X and Y values in the data table. Note that TableCurve offers a root-finding routine in the evaluation procedure of the curve-fit equations as well as in the generated language code output. This may make this X-Y reversal unnecessary.

Restoring Original Data

If you exit the editor with the **Cancel** button, you are asked if you wish to lose all changes made in the editor. If you answer Yes, the original data is fully restored, even if revisions were saved to file within the editor. If you respond No, the changes are incorporated into the data table.

Background Thread Fitting

If **Background Thread Fitting** is active, the fitting will be initiated as soon as the data table is updated upon exit from the editor. If you enter the editor and make no changes or cancel out of the editor, background thread fitting is not affected.

ASCII Editor

The **ASCII Editor** option opens a special TableCurve **ASCII Editor** window designed specifically for editing the program's X-Y data table as an ASCII text file. For editing and copying contents of the various text windows within TableCurve, there is an **ASCII Editor** item in the menu of the specific text window of concern. This data table ASCII Editor option is provided for those who prefer to input or edit a data table as a simple ASCII file using a familiar NotePad-like editor.

Size Limitations

In Windows NT and Windows 2000, this editor should be able to handle any size data table up to TableCurve's 65536 maximum. In systems limited to 16-bit edit controls, such as Windows 95 and Windows 98, the editor is limited to a 64K buffer. On such systems, files imported into the editor larger than 32K are automatically stripped of all whitespace except the necessary delimiter in order to maximize editor capacity. This allows up to about 1500-2000 points.

Automatic Update

When the ASCII Editor is opened, its contents will reflect the current data table. You can edit the data or you can open and edit any ASCII file. When you exit the editor, you will be offered the option of updating the TableCurve data table with the current contents of the editor. If you choose not to save revisions made to the data table, the modified data are automatically saved to **tblcurve.prn**. This file will be listed as the data table's file source in the TableCurve 2D status window.

Background Thread Fitting

If **Background Thread Fitting** is active, the fitting will be initiated as soon as the data table is updated upon exit from the editor. If you enter the editor and make no changes, background thread fitting is not affected.

Generate Data

The **Generate Data** option in the Edit menu and in the main toolbar is used to create data from a mathematical set of equations.

Entering the Function Expression

The function entry uses a simple ASCII multiline editor. The format is similar to that used for **User-Defined Functions**.



You can use the **Cut**, **Copy**, and **Paste** items to move text about or to paste in the formula if you placed it into the clipboard via another program.

All of the functions and constants available within TableCurve 2D can be accessed via a special **Function Insert** help.

You can define as many constants as you want. For example, $\text{SQRT2PI}=\text{SQRT}(2*\text{PI})$ would be defined on one line and used in subsequent lines. Constant expressions are evaluated once and the numeric result is stored. Any assignment to a variable other than **F1-F25** and **Y** is assumed to be a constant. Any expression containing **X** or **XYINDEX** must be assigned either to an **F1-F25** expression or to **Y**. **F1-F25** and **Y** expressions are compiled and are evaluated once for each data point. The Y expression must always be the last line in the signal expression. The following is a simple signal example that generates one random sinusoid. There are five constants and a single Y= signal expression

```
SRATE=5000
NYQ=SRATE/2
AMP=50+50*RANDOM
FREQ=NYQ*(0.01+0.09*RANDOM)
PHASE=2*PI*RANDOM
Y=AMP*SIN(2*PI*X*FREQ+PHASE)
```

The following example generates three random sinusoids and uses F1-F3 functions.

```
SRATE=5000
NYQ=SRATE/2
AMP1=50+50*RANDOM
AMP2=50+50*RANDOM
AMP3=50+50*RANDOM
FREQ1=NYQ*(0.0166667+0.15*RANDOM)
FREQ2=NYQ*(0.1666667+0.1666667*RANDOM);
FREQ3=NYQ*(0.3333333+0.15*RANDOM);
PHASE1=2*PI*RANDOM
PHASE2=2*PI*RANDOM
PHASE3=2*PI*RANDOM
F1=AMP1*SIN(2*PI*X*FREQ1+PHASE1)
F2=AMP2*SIN(2*PI*X*FREQ2+PHASE2)
F3=AMP3*SIN(2*PI*X*FREQ3+PHASE3)
Y=F1+F2+F3
```

Using Derivative Functions

The first derivative function is **DX**(Fn) where n is from 1 to 25 and references an F1 to F25 expression. The second derivative function is **DX2**(Fn). The following example is for the first derivative of a sine function:

```
F1=100*SIN(2*PI*X*5000+PI/2)
```

Y=DX(F1)

Using Integration Functions

The primary integration function is **AIP**(n,start,end,prec). It first seeks to achieve the target precision using a successive step Gaussian Quadrature. If this is unsuccessful, a Romberg procedure follows. If the Romberg fails to achieve the desired precision, an Adaptive Quadrature procedure is used. The following example uses the AIP() function to generate the cumulative of the Log-Normal distribution:

```
LOWER=1E-8
F1=LN($/2.1)/5.2
F2=EXP(-0.5*F1*F1)
Y=100*AIP(F2, LOWER, X, 1E-6)
```

Note that the **\$** symbol is used as the variable of integration. In this example, the second function expression is integrated with **\$** ranging from a lower limit to X, to a precision of 6 significant figures. Infinite limits are supported although you must use the **-INF** and **INF** constants in order to automatically trigger the transforms.

Gaussian Quadrature Functions

The Romberg and Adaptive Quadrature algorithms are generally much slower than the Gaussian Quadrature procedure. For numerically intense integrations, you may wish to restrict the integration to the Gaussian Quadrature procedure. This is done by using the **QIP**(n,start,end,prec) function.

Validation

An expression is extensively validated before it is compiled. If there is a math or parser error, you will be given a clear indication of the error and the cursor will be placed at the location where the validation failed.

Generated X Values

The **X Minimum**, **X Increment**, and **X Maximum** fields are used to create the X values that will be used for the generated signal. The output is limited to 65536 values.

Adding White Noise

To automatically add white noise to the generated data, enter a value in the **Gaussian Noise %** field. Gaussian noise with a standard deviation of this % of the data series standard deviation is added at the conclusion of the signal generation.

Saving a Signal

Files are saved as signals with SIG extensions. Use the **Save** button to save the generated data expression to disk. The expression is always validated before a save is made.

Reading a Signal

Use the **Read** button to read a SIG file from disk. These files are binary and must be created within the program.

TableCurve 2D Graph

When the expression is successfully validated, the generated data is plotted in a **TableCurve 2D Graph**.



The **List Data** option lists the index, x, y values of the generated data. The listing uses the **TableCurve 2D text viewer** facility.



The **Copy Data to Clipboard** option copies the x and y values of the generated data to the clipboard.

Formats include full precision binary (for spreadsheets such as Excel) and ASCII (for pasting into text editors).



The [Save Data to Disk](#) option writes the x and y values of the generated data to a supported file format. These formats include ASCII, Excel 97/2000, Excel 95, Lotus WK3, Lotus WK1, SPSS, or Systat.



Close the graph with the [OK](#) button in order to be offered an option to update the main data table with the generated data.



Close the graph with the [Cancel](#) button to be returned to the expression entry dialog.

Modify Titles

There are two levels of title information. The [Modify Titles](#) option applies only to the basic main, X, and Y titles used for the main data table. This option is used to create or edit these default titles. These titles are saved with the data table and used in most export options.

The main title can be 80 characters in length. The X and Y titles can be up to 40 characters long.

The [Previous](#) button, if enabled, offers the means to use the titles from the prior data set for the current one.

A much more extensive [Custom Titles](#) option is available in all of the program's principal graphs. To use symbols or upper ASCII characters in the main, X, or Y titles appearing within graphs, it is simplest to use this custom titles option.

Copy

The **Copy** option in the main menu copies the current data table to the Windows clipboard as ASCII text, in the Lotus WK1 format, and in the Lotus WK3 format. The ASCII data will contain 15-16 digits of precision

The text format separates the numeric columns with white space rather than tabs. It is used when you paste the data table into a text-based program, such as a word processor. When you paste the data table into Excel, the WK1 format is used. When either Lotus Windows or Quattro Pro Windows is the destination, the full precision WK3 format is used, although only Lotus Windows will preserve TableCurve 2D's 19 digit precision.

This Copy option is limited only by available memory and can copy up to TableCurve 2D's 65536 data table maximum.

Preferences

The **Preferences** option is found within the **Edit** menu and also in the File menu of the **Curve-Fit Graph**. This option is used to set a few global preferences that cannot be automatically set from within the various procedures. Most preferences are saved automatically as each procedure is used and customized.

Numeric Precision in Graph Titles

The **Graph Title Numeric Precision** controls the number of significant figures used in the fit statistic and coefficient titles of the graph. You should use this option to control the number of significant figures in the coefficient and goodness of fit statistics appearing in titles rather than attempting to directly edit the titles using the **Custom Titles** option. Unless you alter the initial characters when modifying these titles, they will be automatically updated for the current equation using this precision preference each time the graph is redrawn.

Numeric Text Display

The **Right Fill Numeric Values** option appends zeros to numeric values if necessary to achieve a specified target length. Otherwise numeric values are truncated at this zero threshold. When checked, the value 19.1 would appear in a table as 19.1000000.

Format for Non-Linear Equations

The **Notational Format for Non-Linear Equations** option offers a compact notational form instead of the full mathematical equation expression for TableCurve's non-linear equations.

Use Bitmaps when Rendering Graphs

The **Use Bitmaps when Rendering Graphs** option specifies that graphs should be first rendered to bitmaps for a smooth rendering to screen. Using bitmaps is recommended, since it represents an aesthetic improvement when stepping through candidate equations. With some video hardware and drivers, however, drawing directly to screen may be slightly faster.

Text Window Type

The **Rich Text Format Windows** option uses RTF-type windows for text display. These windows offer the means to select and copy portions of the overall text, but are considerably slower than TableCurve 2D's standard text windows. For windows whose contents vary based upon the size of a data set, RTF windows are available only for data sets having 128 or fewer points. Data tables larger than 128 points will be displayed using standard text windows regardless of this setting.

Derivative Type

The **Analytic Curve-Fit Equation Derivatives** option specifies that analytic rather than numeric derivatives be used for all derivative computations involving built-in curve-fit equations. These are used for **Curve-Fit graphs** which contain **derivatives on the Y2 axis**, for the computations of derivative minima and maxima that appear in the **Numeric Summary**, and for the derivative values that appear in **Excel**, **Lotus**, **SigmaPlot** export and in the Review's **Copy** option.

In most instances, analytic derivatives are faster. An analytic derivative is also computed to full machine precision (19 significant figures). An analytic derivative computes the actual formula for the first or second derivative.

Typically, a numeric derivative is slower since multiple computations of the function are used to estimate the first or second derivative. A numeric derivative is typically accurate to 8 or more decimal places for a first derivative and to 5 or more decimal places for a second derivative.

Paper Size

Check the [Use A4 as Default Paper Size](#) option if you will be using A4 rather than 8.5 x 11 paper in your printer. This option applies to all [MS Word and RTF file export](#), including the reports generated using the [automation](#) facility.

Main Toolbar

The **Main Toolbar** menu item toggles the main TableCurve 2D tool bar on and off. This tool bar consists of the following options:



Import XY Data Table from File



Append XY Data Table from File



Import XY Data Table from Current Contents of Clipboard



Save Current XY Data Table to File



Enter or Edit XY Data Table via TableCurve Editor



Generate a Data Set from Mathematical Expressions



Enter Basic Titles for XY Data Table



Clear All Current Data



Reset Data to State Prior to Modification



Graphically View XY Data



Graphically Section XY Data



Graphically Compare Data with Imported Reference



Enter and View up to Five Functions of X



Enter a Calculation to Modify XY Data Table



Difference, Cumulative, Normalize



Automated Smoothing



Savitzky-Golay Smoothing



Fourier Denoising



Eigendecomposition Denoising



Fourier Decomposition and Reconstruction



Eigendecomposition and Reconstruction

Process Toolbar

The **Process Toolbar** menu item toggles the processing tool bar on and off. This tool bar consists of the following options:



Spline Estimation



Fourier Estimation



Smoothed Data Spline Estimation



Locally Weighted Regression Spline Estimation



Savitzky-Golay Spline Estimation



Autoregression (AR) Prediction



Set Curve-Fit Preferences for Direct Fit Options



Curve-Fit All Linear Equations, Non-Linear Equations, and UDFs



Curve-Fit All Linear Equations and UDFs



Curve-Fit Simple Equations and UDFs



Curve-Fit Straight Line by Least Squares and by Three Robust Procedures



Curve-Fit All Polynomial Equations, Rational Equations, and UDFs



Curve-Fit Peak Equations and UDFs



Curve-Fit Transition Equations and UDFs



Curve-Fit Kinetic Equations and UDFs



Curve-Fit Waveform Equations and UDFs



Edit Custom Equation Set



Curve-Fit Custom Equation Set and UDFs



Enter or Edit Any or All of TableCurve 2D's 15 UDFs



Curve-Fit Current UDFs



Select Fit Type for Background Thread Fitting



Lights When Background Thread Fitting in Progress (Informational Only)



Begin the TableCurve 2D Review Emphasizing Curve-Fit Graph



Begin the TableCurve 2D Review Emphasizing Equation List

Weighting Data

The data table can be weighted in one of two ways. You may specify a weights column in any **multi-column read operation**, including multi-column ASCII files, Lotus 123 files, Excel files, Quattro Pro files, SigmaPlot files, and dBase files. This is done using the File menu's **Import** option. You may also use the **TableCurve Editor** to enter weights for individual data points. By default all data points have a floating point weight of 1.0.

Weights as Inverse Variances

The data table weights are true floating point multipliers. When each data pair consists of an average of Y observations at a given x , you may wish to set that pair's weight value to the inverse square of the standard deviation. If your weights are true standard deviations, apply a $1/\sigma^2$ conversion in your spreadsheet or enter the weight value followed by $\wedge -2$ in the editor. If you are entering a significant number of data points, you may wish to enter the weight values as standard deviations and apply a $W=1/W^2$ calculation using the main menu's **Calculation** operation.

Normalization of Weights

The weights used in TableCurve 2D are normalized so that the sum of the weights equals the number of active data points. This conserves the degree of freedom relative to unweighted data, and results in coefficient standard errors which better reflect the impact expected from a true floating point weighting scheme. This type of weighting differs from statistical programs which use integer weights to specify the number of identical X,Y pairs. If you are entering identical X,Y pairs and need to see the degree of freedom increased by one for each identical pair, you will have to enter each identical pair separately.

Y-Transformed Equations

The Y-transformed equations use a hidden secondary weighting that is automatically applied in order to produce the best-possible least squares fit. This weighting consists of the inverse square of the derivative of the function of y .

Automatic Weighting Of Replicates

Replicates can be processed either externally or internally.

When **importing** data, the **column selection** dialog offers an **SD Wts** checkbox. When a weights column is imported with this box checked, it is assumed that the imported weights consist of the standard deviations associated with multiple Y measurements at each particular X value. Here, the data table weights will automatically consist of $1/SD^2$. Basic **Error Bars** will then be available in the curve-fit graphs.

The **Process Replicates** option in the **Calculate** menu automatically averages replicate X observations in the data, placing this average in the Y value, and $1/SD^2$ in the weights. This option also internally stores the n for each replicate, enabling **Error Bars** to be specified as Student-t distribution based confidence or prediction intervals.

Data Menu



The **Data** menu offers items for inspecting and modifying the current XY data table:



The **View Data** option in the Data menu offers a simple graph of the data as well as a very fast numeric list facility.



The **View Function(X)** option is used to plot up to five different functions of X.



The **Section Data** option is important for isolating portions of a long data stream for analysis. You can isolate sections graphically by an XY sectioning mode where you box the region of interest, or an X sectioning mode where the left mouse is held down and slid right to disable points, and slid left to enable. You can also numerically specify ranges to include or exclude.



The **Compare with Reference** offers the means to graphically compare the current data set with up to four imported references.

The **Subtract Imported Reference** is used to subtract a reference signal from the current data set. It can be used to remove backgrounds that can be measured separately.



The **Enter Calculation** option is used to enter one or more equations for data transformation.

The **Apply to X-Y Table** is used to apply an existing calculation to the current data table.

The **Cancel Calculation** option clears the calculation from memory.



The **Difference, Cumulative, Normalize** option can difference the data with adjustable order and lag, compute various cumulatives, and normalize for unit area, unit power, unit standard deviation, and zero mean.

The **Process Replicates** procedure is used to average replicates in a data set.

The **XY Statistics** option displays the data table statistics in the status window in a listing that can be saved or printed.

The **Clear Inactive Points** option deletes all inactive points from the current data set.



The **Reset XY Data** will restore modified data back to its original import or generated state.



The **Clear All Data** is used to completely remove all entries from the data table.

View Data

The **View Data** option in the Data menu or the main toolbar offers a **TableCurve 2D graph** of the data as well as a very fast numeric list facility. This option is used only for graphically inspecting the data. In order to toggle individual points on and off, or to activate or deactivate bands of data, you must use the **Section Data** option.

Format settings for this graph will also be used for the graph of the data that is automatically displayed in the status window.

This option may be of interest for data inspection while **Background Thread Fitting** is underway.

Error Bars



Error bars will be available if **replicates** were averaged. If all of the replicates for a given X-value are identical, an SD of 0 is used for the error bar. If only a single value is available, TableCurve 2D uses the conservative approach of setting the SD for that point to range of the Y-variable.

List



The **List Data** option lists the index, x, y values in a three column table. The listing uses the **TableCurve 2D text viewer** facility.

Copy



The **Copy Data to Clipboard** option copies the x and y values to the clipboard. Formats include full precision binary (for spreadsheets such as Excel) and ASCII (for pasting into text editors). Note that the current data table can also be copied to the clipboard using the **Copy** option in the main Edit menu.

Save



The **Save Data to Disk** option writes the x and y values to a supported file format. These formats include ASCII, Excel 97, Excel 95, Lotus WK3, Lotus WK1, SPSS, or Systat. Note that the current data table can also be saved to disk using the **Save As** option in the main File menu.

MS Word/RTF Export



The **MS Word/RTF File Export** option is used to save the current graph to either a MS Word file or a portable RTF (Rich Text Format) file. The graph is inserted into the file as a Windows metafile.

Function(X)

This option is used to graph up to five functions of X across the X range of your choice. The graph can be scaled, formatted, printed, or copied to the clipboard. You may also plot the derivatives or cumulative area of the functions and you may numerically evaluate a function you have entered.

Function(X) Entry

Enter up to five functions of X you wish to graph. You may enter any numeric calculation. All of the functions and constants available within TableCurve 2D can be accessed via a special [Function Insert](#) help. The Y= prefixes are automatically supplied. The Function(X) functions are limited to single line expressions. Use the Copy, Cut, and Paste buttons to paste in expressions from the clipboard, or to copy or modify the function text. If the functions are not successfully validated, the math or parser error causing the failure will be shown and the cursor will be placed at the location in the function where the error was detected.

There are four format options for displaying the Function(X) functions:

- **Fn** - displays the functions in a single plot graph
- **Fn, D1** - displays the functions (Y axis) and first derivatives (Y2 axis) in a dual plot graph
- **Fn, D2** - displays the functions (Y axis) and second derivatives (Y2 axis) in a dual plot graph
- **Fn, Int** - displays the functions (Y axis) and cumulative areas (Y2 axis) in a dual plot graph

Save

Use the [Save](#) button to save the view functions(x) to disk. Although these VFX files are simple ASCII files, it is recommended that they be created only from within TableCurve 2D.

Read

Use the [Read](#) button to import a VFX file.

View Function(X) Graph

Simply select one of the four format options to create the desired [TableCurve 2D Graph](#).

Note that cumulative areas require the evaluation of an integral for every pixel drawn in the plot. With a number of functions, with mathematically complex functions, or with large high-resolution graphs, these integrations may require some time. The integrations are done from a lower limit which corresponds with the lower limit of the displayed graph. By changing the lower X of the graph, you are thus changing this lower limit of integration.

The functions will be plotted in colors which correspond with the component colors. These colors will also be used for the main titles of the [TableCurve 2D Graph](#).

Undo



Use the [Undo](#) item to return to the Function entry screen in order to revise the functions or X range.

Zoom

Ten levels of zoom-out are available by using the scrollbar in the graph's control panel. If this is a [Fn,Int](#)

graph which displays cumulative areas, there may be an appreciable delay in updating the graph following a change in the zoom setting.

Evaluation

 Eval. The **Eval** item is used to perform a full-featured numeric evaluation of one of the functions. If two or more functions are present, you must choose one for this numeric evaluation.

Section Data

This option is used to section the data into active and inactive regions, or to disable or enable specific data points. Only active points are fitted.

The option will present a dual coupled [TableCurve 2D Graph](#). The upper graph will be scaled for only the active points, those used within curve-fitting. The lower graph will be scaled for all data points. You may graphically section either graph, and the changes will be reflected in both. Inactive points are shown in the current inactive point color. You may section for active or included bands, or inactive or excluded bands, in one of two ways.

Numeric Sectioning

When you know exactly which ranges to include or exclude, numeric sectioning may be simplest. The sectioning can be set to either [Include](#) or [Exclude](#) the points in the specified region. A sectioning region consists of 2D box which is defined by an initial X [X_i], a final X [X_f], an initial Y [Y_i], and a final Y [Y_f]. If no entries are made for a given variable, the program assumes the full range of the variable when including points, and no range on the variable when excluding points. To apply the numeric sectioning to the data, you must select either the [Apply Existing](#) or [Apply New](#) button. The Apply Existing applies the given region to the current status of the points whereas the Apply New option resets all points prior to the sectioning.



Graphical X-Sectioning

It is often simpler and faster to section graphically. When the [X Sectioning](#) mode is selected, the mouse is used to select the x range of interest. To exclude points, place the mouse cursor at the initial point to be excluded, click and hold the left mouse button down, and slide the mouse to the right. As you do so, the points will be grayed (indicating that they have been deactivated or excluded). Once the left mouse button is released, both graphs will reflect the sectioning. To reinstate a band of points, start at the rightmost point desired, click and hold the left mouse button down, and slide the mouse to the left. As you do so, the points will be restored to an active state.



Graphical X-Y Sectioning

There may be times where the data of interest are defined by bounds in both x and y. When the [XY Sectioning](#) mode is selected, the mouse is used in the upper graph to enclose the x-y region of interest. To include the region that is to be graphically defined, click and hold down the left mouse button at one corner of the x-y rectangle in the upper graph and drag the rectangle so that it encloses the region desired. To exclude the region that is to be graphically defined, the same graphical procedure is followed except that the right (rather than the left) mouse button is used.



Graphical Toggling of Points

To toggle a single point between active and inactive, the [TableCurve 2D Graph](#) must be in [XY Sectioning](#), [X Sectioning](#), or [Process Points](#) mode. In Process Points mode, the cursor will track points, and only horizontal mouse movement is processed. In either mode, simply place the mouse cursor on the point desired. The mouse cursor will change to a crosshair with a point in its center. If you have the hints active, the hint color will change and indicate the point under the cursor. If the graph's status bar is active, you will see the point under the cursor also indicated there. Click the left mouse button without moving the mouse to toggle the point on or off.



Moving Data Points

While not normally recommended, TableCurve 2D allows you to move any point to a different Y-value. This must be done in **Process Points** mode. Moving a single data point may be one way to deal with single channel artifacts. Simply place the mouse cursor on the point desired, hold the left mouse button down, and move the cursor up or down to the new Y-location. The first time this is done, you will be asked to confirm that you genuinely want to alter the value of the data point in this manner.

Zoom-Mode

In cases of highly concentrated data, you may wish to zoom-in a given region prior to sectioning. The zoom-mode is provided for this purpose. Either graph can be zoomed in this mode by clicking and holding down the left mouse button while enclosing the zoom-in region desired.

To restore normal scaling, click the right mouse button anywhere within the graph region.

Clear Entries



The **Clear** button restores all data points to an active state and clears the entry fields. Points that have been moved to a new Y-location are not reset.

Automation



The **Production Facility** button is used to initiate an automated processing where any number of data sets can be automatically sectioned in an unattended manner. This option offers only Excel output.



The **Reset** button restores the data to its state when first entering the procedure. If an **Automation Session** is in progress, the **Reset** button can be used to terminate the automated processing.

Accepting the Sectioning

Exit the dialog with **OK** to accept the sectioning shown in the graph, or **Cancel** to exit without adopting any modifications.

Compare with Reference

This option is useful if you want to compare the current data set with up to four reference data sets. A reference can come from any of the supported **import formats**, including equation fit references that have been saved to disk. When an equation reference file is selected for data import, discrete data are generated. You must supply a starting and ending X-value and also either an X-increment or total data count.

A reference can represent a single data file or a set of appended data files produced by repeated **Append** operations followed by a final **Save As**. A discrete data reference can be generated from a curve fit using the Review's **Eval** or any **Export** option; a continuous equation reference can also be saved to disk. Such a fit can likewise be based on a single set, or any number of appended sets intended to produce a standard.

The data sets are displayed in a **TableCurve 2D Graph**.

Adding References

Use the buttons labeled **1**, **2**, **3**, and **4** to add a reference.

Removing References



Use the **Reset** button to remove references in the reverse of the order added.

MS Word/RTF Export



The **MS Word/RTF File Export** option is used to save the current graph to either a MS Word file or a portable RTF (Rich Text Format) file. The graph is inserted into the file as a Windows metafile.

Subtract Imported Reference

This option is used to subtract a reference data set, having matching X values, from the current data set. This is generally used to subtract a baseline or background, or for fitting the difference between two different data sets. The reference can be imported from any of TableCurve 2D's [supported file formats](#).

The reference is subtracted and a [TableCurve 2D Graph](#) displays the subtracted data. Click on [OK](#) to accept the correction and modify the data table, or click on [Undo](#) to restore the original uncorrected data.

If the opposite data set is being subtracted, apply a $Y=-Y$ [Calculation](#) to change the sign of the result.

If [Background Thread Fitting](#) is active, the fitting will be initiated as soon as the subtraction is made. The fitting will thus be taking place in the background when you are inspecting the result of the modification.

Enter Calculation

The **Enter Calculation** in the **Data menu** offers the means to transform one or more of the data table variables.

A calculation consists of an X= expression, a Y=expression, and a W= expression. Only one expression need be present in a calculation. You may enter any numeric calculation. All of the functions and constants available within TableCurve can be accessed via a special **Function Insert** help. The X=, Y=, and W= prefixes are automatically supplied. When entering a calculation with multiple cross references to X and Y, bear in mind that the X calculation will be applied first, then the Y, and finally the weight calculation.

Calculation Types

A calculation can be used to change the existing variable. As examples, the expression $X=X*2.54$ converts X-values in inches to centimeters and the expression $Y=(4/3)*PI*Y^3$ converts Y-values representing a radius to a volume. A calculation can also be used to calculate one variable based upon another. As examples, the expression $Y=Y+1.232+2.323*X+3.3433*X^2+4.3234*X^3$ increments the original Y variable by the value of a polynomial in X. The expression $W=1/Y$ assigns **weights** equal to the inverse of the Y-values.

Copy, Cut, Paste

Use these buttons to paste in expressions from the clipboard, or to modify the text across the various fields.

Read

Calculations that have been saved to disk are read by using the Read button. It is a good idea to save a calculation that will likely be used on future data sets. Calculation files are binary files with .CLC extensions. They can be created only within the program.

Save

The calculation active at the close of any given session is not automatically saved across sessions. You must explicitly save a calculation to disk using the Save button in order to have it available in a future session. Calculation files have .CLC extensions.

Validation

The calculation will be validated when you exit the Calculation entry screen. If there is an error in any of the expressions, an error message will report a specific parser or math error and you will be shown where within the expression the evaluation failed. The cursor will also be placed at this position. Note that a calculation must be defined at the Xmean and/or Ymean of the data table or at values of 1.0 if the data table is empty. Calculations are compiled for a very rapid processing with large data sets.

Option to Apply

You will be given an option to immediately **Apply the Calculation** you have entered to the current data table. Answer no if the calculation is to be applied only to additional data that will be appended to the current table.

If **Background Thread Fitting** is active, the fitting will be initiated as soon as the calculation is applied. The fitting will thus be taking place in the background when you are inspecting the result of the calculation in

the TableCurve 2D Graph.

Apply to X-Y Table

This option applies the current **calculation** to the full X-Y data table. The X-calculation, if present, is first applied to the full data table, then the Y-calculation, if present, is applied to the full table. Finally, the weight calculation is applied, if present, again to the full table. If you have an X calculation that references Y, and a Y calculation that references X, note that the X calculation will use the original Y and the Y calculation will use the modified X.

Undo

After the calculation is applied, you are presented with a **TableCurve 2D Graph** that graphically renders the data after the calculation has been applied. You can undo the calculation if you are not satisfied with the results or if you want to experiment with different calculations. If you choose **Undo**, the original data will be restored. Choose **OK** to accept the modifications.

If **Background Thread Fitting** is active, the fitting will be initiated as soon as the calculation is applied. The fitting will thus be taking place in the background when you are inspecting the result of the calculation in the **TableCurve 2D Graph**.

Cancel Calculation

Use this [Cancel Calculation](#) selection to delete the current Calculation from memory. If you want to use this same calculation in a future session, first save it to disk by using the Save item in the [Enter Calculation](#) option.

Difference, Cumulative, Normalize

The **Difference, Cumulative, Normalize** option in the Data menu or the main toolbar can difference the data with adjustable order and lag, compute various cumulatives, and normalize for unit area, unit power, unit standard deviation, and zero mean.

This option presents a **TableCurve 2D graph** displaying the input and detrended data.

Process

The procedure can generate normalized **Data**. The **Difference** item specifies that differencing, the discrete analog of differentiation, is to take place. The **Cumulative** option specifies that a cumulative, the discrete analog of integration, will be generated.

Diff. Options

The difference options are only available if **Difference** is checked. The **Order** field specifies the order of the differencing. Most often first order differencing is used. This is the equivalent of taking the first derivative of continuous data. Orders through 4 are supported. The **Lag** field specifies the delta of the differencing. Usually a lag 1 is used to produce the difference from adjacent elements. A lag of 12 is used sometimes to seasonally adjust monthly data. Similarly, a lag of 4 can be used to seasonally adjust quarterly data.

If **dY** is selected, the differencing is based only on the Y-values in the data set. If **dY/dX** is selected, the differencing is processed like a derivative generating a discrete rate value.

Cum. Options

The cumulative options are only available if **Cumulative** is checked. The **Simple** option generates a cumulative by basic addition. The **Area** option uses the x-spacing to produce an area cumulative. The **Power** option uses the x-spacing to produce a TISA (time-integral squared amplitude) cumulative power data series. For oscillatory data, the cumulative power curve will be continuously increasing.

Normalize

The normalizations apply to all processing options. No normalization is made if **None** is checked. The data can be normalized to **Unit Area**, **Unit Power** (TISA), or **Unit SD** (standard deviation). If the **Zero Mean** box is also checked, the mean of the resulting series will be zero.

Data Count

If differencing is done, the output data count will be reduced. The input and output counts are displayed in the **nIn** and **nOut** informational fields.

List



The **List Data** option lists the index, x-values, and the processed data. The listing uses the **TableCurve 2D text viewer** facility.

Copy



The **Copy Data to Clipboard** option copies the x values and processed data to the clipboard. Formats include full precision binary (for spreadsheets such as Excel) and ASCII (for pasting into text editors).

Save



The **Save Data to Disk** option writes the x-values and processed data to a supported file format.

These formats include ASCII, Excel 97/2000, Excel 95, Lotus WK3, Lotus WK1, SPSS, or Systat.

Production Facility



The **TableCurve 2D Automation** facility allows unattended processing of large numbers of data sets. The data sets can be consolidated in an Excel file or acquired using a DLL. The graphs can be exported to a Microsoft Word/RTF file, while the processed data can be exported to an Excel 95 or Excel 97/2000 file.



The **Reset** button restores the data to its state when first entering the procedure. If an **Automation Session** is in progress, the **Reset** button can be used to terminate the automated processing.

MS Word/RTF Export



The **MS Word/RTF File Export** option is used to save the current graph to either a MS Word file or a portable RTF (Rich Text Format) file. The graph is inserted into the file as a Windows metafile.

Updating the Data Table



When exiting this procedure with the **OK** button, an option will be presented to update TableCurve 2D's main data table with the processed data.

Process Replicates

The **Process Replicates** option in the **Data menu** is the preferred procedure for managing replicates in TableCurve 2D. This option automatically searches a composite data table for replicate X values, and replaces the various instances with a single table entry where the Y value consists of the mean of the individual Y values, and the weight value consists of $1/\text{variance}$ ($1/\text{SD}^2$). This option also stores the n for each replicate entry, enabling **Error Bars** to be expressed as Student-t distribution based confidence or prediction intervals. In TableCurve 2D, replicates are based on the X-values coinciding to within 1E-8 fractional precision.

Saving

If you choose to **Save** a processed replicate data table, you should do so as an ASCII file in order to preserve the n for each of the entries.

Appending

The recommended approach for replicates in TableCurve 2D is to externally create a composite data set containing the multiple experiments, or to use the **Append** options within TableCurve 2D to create a single data table containing all pertinent replicates. It is recommended that you save this file as the parent for all subsequent work. This composite table can then be automatically processed for replicates using this option.

If you are appending multiple data sets, you should not invoke this option until you have appended all of the data sets. TableCurve 2D's data table retains only the variance at each X-value, not the number of samples at that X. This means that you cannot meaningfully append a data set to a table that already consists of averaged values since the averaged set and the additional set would be given equal weight in the averaging.

Note that **Column-Specified Import** does allow the importing of **weights** as standard deviations, but the n for each entry is not available, and as such, **Error Bars** associated with replicates are limited to multiples of the standard deviation and to confidence intervals based on a normal distribution. Student-t distribution based confidence and prediction intervals will not be available.

Restoring Original Data



When the replicates are averaged, whether by this option or automatically, the original appended series can be restored with the **Reset XY Data** option.

Error Bars



Error bars will be available in the Data menu's **View Data** option if replicates are averaged. If all of the replicates for a given X-value are identical, an SD of 0 is used for the error bar. If only a single value is available, TableCurve 2D uses the conservative approach of setting the SD for that point to the range of the Y-variable.

XY Statistics

The [XY Statistics option](#) opens a [TableCurve text window](#) which lists the basic data table statistics shown in the status window. You may print this data, save it to text or spreadsheet files, copy it to the clipboard in both text and spreadsheet format, or edit it using TableCurve's [ASCII editor](#).

Clear Inactive Points

This option clears all points in the current X-Y data set that have been excluded from fitting. Such points are marked as inactive either in the [TableCurve Editor](#) or [ASCII Editor](#) or by [Sectioning](#) of the X-Y Data.

When a high percentage of points have been marked as inactive, clearing these points prior to fitting can result in faster curve-fit processing, especially with non-linear equations. After discarding the inactive points, you may want to use the File menu's [Save As](#) option to save the data absent these excluded points.

Reset XY Data

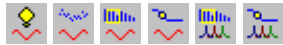
The **Reset XY Data** option in the **Data menu** or the **main toolbar** will restore modified data back to the original import or generated state. Many of the procedures in TableCurve 2D offer the option of replacing the current data table with a modified data set. This option will immediately restore the data to the initial state. Note that if successive revisions are made, this option discards all of them.



Clear All Data

This option clears all entries from the current data table.

Filter Menu



The **Filter** menu offers signal processing procedures that modify the data table by smoothing, removing noise, and isolating components. These include advanced processing procedures in the Fourier and Eigendecomposition domains.



The **Automated Smoothing** option simplifies the use of six different procedures for smoothing the data table.



The **Savitzky-Golay Smoothing** procedure offers effective time-domain smoothing for data sets with uniform X-spacing. The algorithm offers adjustable order (quartic is typical), automatic sequential passes (three is about optimum), and optional first through eighth smoothed derivatives.



The **Fourier Denoising** option is a specialized Fourier filtration procedure that sets either a frequency threshold for low pass frequency-domain filtration, or a signal threshold for zeroing all spectral elements below a given power. The time domain data are reconstructed using the inverse FFT.



The **Eigendecomposition Denoising** option accomplishes a similar function except that the filtration occurs by zeroing those eigenmodes that contain noise. By using a high order decomposition, it is often possible to remove nearly all of the noise within a signal.



The **Fourier Filtering** option is an extensive Fourier domain filtering and component isolation procedure. This procedure supports data tapering windows so that low power components can be isolated and reconstructed.



The **Eigendecomposition Filtering** option offers full eigenmode filtering and reconstruction. Eigendecomposition partitions by signal strength rather than by frequency. In addition to the data, the reconstruction can optionally consist of the eigenvectors, the principal components, the data components, FFTs of the data components, or an FFT spectrum of the data.

Automated Smoothing

The [Automated Smoothing](#) option is used to smooth noisy data with minimal user intervention. This procedure is useful for rapidly comparing the six different smoothing methods offered by TableCurve 2D. The raw and smoothed data are shown in a [TableCurve 2D Graph](#) for rapid determination of the effectiveness of smoothing.

There are six methods of smoothing available:

- [FFT Filtering](#)
- [Loess](#)
- [Gaussian Convolution](#)
- [Savitzky-Golay Filtering](#)
- [Eigendecomposition Thresholding](#)
- [Kaiser-Bessel Lowpass Filtering](#)

Automatic Uniform X-Spacing

Only the [Loess](#) algorithm is designed to manage data with non-uniform X-spacing. The other algorithms perform best with constant X-spacing.

This automated smoothing procedure fully manages the constant X-spacing required by these algorithms. If uniformly-spaced X-values are not present, this procedure will first fit a [constrained cubic spline](#) and generate a new data stream containing the same data count. The uniform data are then processed by all of the algorithms except Loess, where it is not needed.

Optimizing Noise Removal and Derivatives

With only a single smoothing level adjustment, this automated smoothing procedure usually offers a close to optimum level of noise removal for each of these algorithms. Distortion of the underlying data trends will not usually occur until high smoothing levels are specified. For instances where an optimum noise removal is sought, TableCurve 2D offers separate Savitzky-Golay, Fourier, and Eigendecomposition procedures.

This optimization of the algorithm may be especially critical with derivatives. The derivatives available in this automated smoothing option are based on derivatives of a cubic spline interpolant of the smooth data. In general, for noisy data, derivatives based directly on the [Savitzky-Golay filter](#) are more accurate, especially beyond the first order derivative.

When derivatives are not the focus and you want the maximum in noise removal without impacting the underlying signal, consider optimizing the [Eigendecomposition Denoising](#) procedure. This can be especially effective for data sets with several hundred or more points in which a high model order can be specified. In this instance, a large number of eigenmodes can fully capture the noise component.



The separate [Savitzky-Golay Smoothing](#) procedure offers much greater control of the Savitzky-Golay algorithm and smooth derivatives based directly on the smoothing filter.



The [Savitzky-Golay Spline Estimation](#) option additionally adds interpolation.



The [Local-Regression Spline Estimation](#) option offers a much greater control of the Loess algorithm, including higher orders, and also adds interpolation.



The **Fourier Denoising** option offers both frequency and signal thresholding for low pass frequency-domain filtration as well as raised data tapers to reduce spectral leakage.



The **Fourier Filtering** option additionally offers channel by channel control of the Fourier domain filtering.



The **Eigendecomposition Denoising** option offers a greater control of the algorithm and the means to visually separate the signal and noise eigenmodes via singular value magnitudes.



The **Eigendecomposition Filtering** option additionally offers the means to visualize individual eigenmodes both in the time and frequency domains. This can help determine the optimum signal-noise transition.

Level (% Smoothing)

The % smoothing for each of the algorithms defines the breadth of a smoothing or filtering window.

For the **FFT Filtering**, the % smoothing controls the channels that are zeroed in the frequency domain. A 10% smoothing level zeros the upper 1/2 of the frequency channels. A value of 20% zeros the upper 3/4 of the channels. A value of 50 zeros the upper 9/10 of the channels. For peak-type data, the optimum smoothing level is generally between 25 and 40%.

For the **Loess** and **Savitzky-Golay** procedures, the % smoothing determines the actual smoothing window. A 10% smoothing level results in a smoothing window containing 10% of the data points.

For the **Gaussian Convolution** smoothing, the % smoothing corresponds with the Gaussian FWHM being convolved with the data. A 5% smoothing levels convolves a Gaussian having a FWHM 5 times the average X spacing (sampling interval) of the data set. The Gaussian convolution also includes an intrinsic frequency domain filtration which is automatically determined.

The **Eigendecomposition** % smoothing specifies the eigenmode count attributed to noise and discarded. If the data count is sufficient, a model order 40 eigendecomposition is made. A smoothing level of 36% thus retains 4 eigenmodes as signal and discards 36 eigenmodes as noise.

The **Kaiser-Bessel** % smoothing corresponds with the normalized cutoff frequency of the filter. 0% corresponds with a frequency cutoff of 1.0, 25% with 0.1, 50% with 0.01, and 75% with 0.001.

Modify the % smoothing by entering the desired value; by changing the value with the up or down spin buttons; or holding down the right mouse button on the edit field and spin controls, selecting the level from the popup menu. After a brief delay, the smoothed data will be reflected within the graph.

AI Expert



To make the determination of smoothing levels as automatic as possible, TableCurve 2D offers an **AI Expert** option which seeks to automatically determine the optimum smoothing level. This is that level of smoothing which offers the greatest possible noise reduction without adversely affecting features within the data.

Depending on the algorithm, the AI Expert is based upon a first derivative of the overall standard error (between raw and smoothed) relative to smoothing power, a first derivative of the noise reduction relative to smoothing power, or an estimation of the optimum **eigendecomposition transition**.

The AI Expert option will work reasonably well for many data sets. It is most easily fooled on data sets

with a small number of points, and on data sets in which the noise is not consistent across the X range of the data.

Derivatives

First and second derivatives can be generated instead of zero-order smooth data. For all of the smoothing algorithms, including Savitzky-Golay, these derivatives are computed from a cubic spline interpolant of the zero-order smoothed data. While these derivatives are often very stable, the most accurate derivatives will usually come from creating a derivative filter with the separate [Savitzky-Golay](#) procedure.

FFT Filtering

This is essentially the automated version of what you can do manually in the [Fourier Smoothing and Denoising](#) and [Fourier Filtering and Reconstruction](#) options.

This option removes any linear trend which might appear as a low frequency component in the FFT, performs a forward [Best Exact-N FFT](#), zeros the higher frequency components, performs an inverse [Best Exact-N FFT](#), restores the linear trend, and presents the smoothed data, all in a single automated step.

Because the signal tends to appear only at low frequencies, TableCurve 2D uses the aforementioned non-linear smoothing scale.

Because of the effectiveness of the FFT, this algorithm is quite fast with large data sets.

In lowpass filtering, low frequency harmonic oscillations that are present will pass without being filtered. If you see unwanted sinusoidal behavior within the baseline areas, a time domain procedure (Savitzky-Golay or Loess) should probably be used.

Loess

The Loess procedure is a locally weighted regression smoothing algorithm that performs a full least-squares fit for each data point. Its strength rests in its ability to directly manage data with non-uniformly spaced X values. This is most numerically intense of the smoothing algorithms and very slow with large data sets.

The Loess smoothing implemented in this automated procedure fits a linear model. In general, this means there will be some attenuation of data features at higher smoothing levels.

A greater control of the Loess procedure, including higher model orders, is available in the [Local Regression Estimation](#) option.

Gaussian Convolution

This algorithm is currently unique to TableCurve 2D. It uses an automatic [Best Exact-N FFT](#) filtering for global smoothing, and convolves a narrow width Gaussian for local smoothing.

In the Gaussian Convolution smoothing, the data are transformed to the frequency domain by a forward [Best Exact-N FFT](#), and there a Gaussian response function is convolved with the data. The frequency domain data are then automatically filtered, and the inverse [Best Exact-N FFT](#) is made.

The frequency domain filtering is automatically determined with this algorithm, and it will have some dependency on the Gaussian width used in the convolution. The % smoothing specifies only the FWHM of the Gaussian convolving the data, this as a multiple of the average X-spacing in the data set. As such, this algorithm will generally require % smoothing levels between 2 and 5, this corresponding with

Gaussian response function FWHM of 2x to 5x, the average sampling interval. Anything greater is likely to produce unwanted attenuation of the data features. A response function convolution procedure does conserve overall area, however, so such attenuation is not as harmful as might be the case in a time domain procedure.

This algorithm can produce a very high degree of noise reduction, but bears the same limitations as the FFT Filtering. Since the FFT procedures are quite fast, the algorithm is efficient with large data sets, though somewhat slower than the simple FFT Filtering.

Savitzky-Golay

This time-domain method of smoothing is based on least squares quartic polynomial fitting across a moving window within the data. The method was originally designed to preserve the higher moments within spectral data. The algorithm has been modified by TableCurve 2D and will offer a higher level smoothing than that traditionally associated with Savitzky-Golay. The TableCurve 2D implementation of the Savitzky-Golay algorithm employs sequential internal smoothing passes to improve overall noise reduction.

The higher order polynomial model makes it possible to achieve a high level of smoothing without attenuation of data features. Because this algorithm relies on the linearity of an unweighted polynomial model, it is also reasonably efficient with large data sets.

Eigendecomposition

This is essentially the automated version of what you can do manually in the [Eigendecomposition Denoising](#) and [Eigendecomposition Filtering](#) options.

The smoothing occurs by performing an [eigendecomposition](#) and discarding some number of the least-ranked eigenmodes, and reconstructing the data using only the highest-ranked eigenmodes. If the data table size permits, a matrix order 40 decomposition is made. Otherwise a smaller model order is used.

Since the eigendecomposition uses [SVD](#), it is a slower procedure than FFT filtering.

When a clear differentiation is possible between signal-bearing eigenmodes and noise-bearing ones, this algorithm offers the highest level of smoothing with the least distortion of underlying data. Unlike the other methods, it can potentially remove the entirety of random noise from a signal without touching the deterministic component.

Kaiser-Bessel

This option applies a Kaiser-Bessel time-domain lowpass digital filter to the data stream. The results are similar to the Fourier smoothing, except that a gentler transition occurs in the frequency domain.

The low pass filter's sidelobe level is set to about -60 dB and the cutoff frequency is determined by the smoothing level.

Applying a time-domain digital filter is an exceedingly fast operation, even with large data sets.

Equivalent Noise %

TableCurve 2D offers a robust noise estimation procedure that may be of some value for low-frequency signals. A cubic polynomial interpolation is made for each point using the two points to the left and the two to the right (excluding the current point). The difference between the interpolated and data values is used to generate a measure of the white noise present. This assumes that the data can be locally characterized by a smooth cubic interpolant.

The **Raw Data** value reports the estimated white noise in the incoming data; the **Smoothed** value reports the estimated white noise for the smoothed data. The percent is given as the amount of estimated noise remaining after smoothing.

This estimate does not serve as an indicator for oversmoothing. In general, oversmoothing is easily observed visually in the attenuation of data features.

The **r-squared correlation coefficient** is also reported. An r^2 of 1 is a perfect correlation while a value of 0 means the smoothed and unsmoothed data are completely uncorrelated.

List



The **List Data** option lists the index, x-values, and the processed data. The listing uses the **TableCurve 2D text viewer** facility.

Copy



The **Copy Data to Clipboard** option copies the x-values and processed data to the clipboard. Formats include full precision binary (for spreadsheets such as Excel) and ASCII (for pasting into text editors).

Save



The **Save Data to Disk** option writes the x-values and processed data to a supported file format. These formats include ASCII, Excel 97/2000, Excel 95, Lotus WK3, Lotus WK1, SPSS, or Systat.

Residuals



The **Residuals** button is used to graphically display the difference between the original and smoothed data.

Production Facility



The **TableCurve 2D Automation** facility allows unattended processing of large numbers of data sets. The data sets can be consolidated in an Excel file or acquired using a DLL. The graphs and reports can be exported to a Microsoft Word/RTF file, while the processed data can be exported to an Excel 95 or Excel 97/2000 file.



The **Reset** button restores the data to its state when first entering the procedure. If an **Automation Session** is in progress, the **Reset** button can be used to terminate the automated processing.

MS Word/RTF Export



The **MS Word/RTF File Export** option is used to save the current graph to either a MS Word file or a portable RTF (Rich Text Format) file. The graph is inserted into the file as a Windows metafile.

Automatic Update

The List and Residuals windows are automatically updated when any change is made to the algorithm's settings.

Updating the Data Table



When exiting this procedure with the **OK** button, an option will be presented to update TableCurve

2D's main data table with the processed data.

If **Background Thread Fitting** is active, the fitting will be initiated as soon as the smoothing is accepted and the data are rescanned.



Savitzky-Golay Smoothing

The **Savitzky-Golay Smoothing** procedure in the Filter menu or the main toolbar offers effective time-domain smoothing for data sets with uniform X-spacing. The algorithm offers adjustable order, automatic sequential passes, and optional first through eighth smoothed derivatives.

Uniformly spaced data are needed so that no additional noise is introduced. This procedure does not automatically generate uniform data for the Savitzky-Golay algorithm. If needed, the **Spline Estimation** or **Local Regression Spline Estimation** option can be used to create an interpolated data set with constant X-spacing.

The raw and smoothed data are shown in a **TableCurve 2D Graph** for rapid determination of the effectiveness of smoothing.

Savitzky-Golay

This time-domain method of smoothing is based on least squares polynomial fitting across a moving window within the data. The method was originally designed to preserve the higher moments within time-domain spectral peak data. The TableCurve 2D implementation of the Savitzky-Golay algorithm offers sequential internal smoothing passes to improve overall noise reduction.

To generate a fourth derivative, a minimum quartic (order 4) smoothing must be used. For a third derivative, a minimum cubic (order 3) smoothing is needed. Similarly, a second derivative requires a minimum quadratic (order 2) smoothing.

A higher order polynomial makes it possible to achieve a high level of smoothing without attenuating the extrema in the data. Because this algorithm relies on the linearity of an unweighted polynomial model, it is also quite efficient with large data sets.

The Savitzky-Golay procedure requires a constant X-spacing within the data. If needed, use one of the options in the **Estimate menu** to generate uniformly spaced data.

Options

The **Win n** field specifies the one-sided width of the moving window. The smoothness will increase as the size of the window becomes greater.

The **Order** specifies the polynomial model order. A quadratic (order 2) is the minimum. Orders through 12 are supported. Typically, Savitzky-Golay smoothing uses a quartic (order 4) polynomial.

The **Passes** field specifies the number of sequential applications of the smoothing filter. In general, three passes is about optimum.

The **Zero Edges** option is useful for derivatives when edge effects are adversely corrupting the computation of the derivative. When this box is checked, all points that are in this zone of edge effects are zeroed. Note that increasing the pass count expands this edge effect zone.

Process

The Savitzky-Golay procedure can smooth to a derivative. You can choose to simply smooth the **Data**, or you can smooth to a derivative. Derivatives 1 through 8 are available. You cannot select a derivative order higher than the polynomial model order currently set.

Estimated Noise Reduction

TableCurve 2D offers a robust noise estimation procedure that may be of some value for low-frequency

signals. A cubic polynomial interpolation is made for each point using the two points to the left and the two to the right (excluding the current point). The difference between the interpolated and data values is used to generate a measure of the white noise present. This assumes that the data can be locally characterized by a smooth cubic interpolant.

The **Raw** value reports the estimated white noise in the incoming data, the **Smooth** value the estimated white noise for the smoothed data. The percent is given as the amount of estimated noise remaining after smoothing.

The **r-squared correlation coefficient** is also reported. An r^2 of 1 is a perfect correlation while a value of 0 means the smoothed and unsmoothed data are completely uncorrelated.

List



The **List Data** option lists the index, x-values, and the smoothed data. The listing uses the **TableCurve 2D text viewer** facility.

Copy



The **Copy Data to Clipboard** option copies the x values and smoothed data to the clipboard. Formats include full precision binary (for spreadsheets such as Excel) and ASCII (for pasting into text editors).

Save



The **Save Data to Disk** option writes the x-values and smoothed data to a supported file format. These formats include ASCII, Excel 97/2000, Excel 95, Lotus WK3, Lotus WK1, SPSS, or Systat.

Production Facility



The **TableCurve 2D Automation** facility allows unattended processing of large numbers of data sets. The data sets can be consolidated in an Excel file or acquired using a DLL. The graphs can be exported to an MS Word/RTF file, while the processed data can be exported to an Excel 95 or Excel 97/2000 file.



The **Reset** button restores the data to its state when first entering the procedure. If an **Automation Session** is in progress, the **Reset** button can be used to terminate the automated processing.

Residuals



The **Residuals** button is used to graphically display the difference between the original and smoothed data.

Numeric Summary



The **Numeric Summary** displays a basic summary of the analysis and a listing of the Savitzky-Golay filter coefficients. If a derivative is currently selected, both the zero order data and the specified derivative coefficients are listed.

MS Word/RTF Export



The **MS Word/RTF File Export** option is used to save the current graphs and reports to either a MS Word file or a portable RTF (Rich Text Format) file. The graphs are inserted into the file as Windows metafiles.

Automatic Update

The List, Residuals, and Numeric Summary windows are automatically updated when any changes are made in the algorithm's settings.

Updating the Data Table



When exiting this procedure with the **OK** button, an option will be presented to update TableCurve 2D's main data table with the smoothed data.

If **Background Thread Fitting** is active, the fitting will be initiated as soon as the smoothing is accepted and the data is rescanned.



Fourier Denoising

The **Fourier Denoising** option in the Filter menu or the main toolbar is a specialized Fourier filtration procedure that sets either a frequency threshold for low pass frequency-domain filtration, or a signal threshold for zeroing all spectral elements below a given dB (normalized) value. This option is used exclusively to remove noise via a Fourier decomposition.

Uniformly spaced data are needed so that no additional noise is introduced. This procedure does not automatically generate uniform data for the Fourier decomposition. If needed, the **Spline Estimation** or **Local Regression Spline Estimation** option can be used to create an interpolated data set with constant X-spacing.

The option presents a dual TableCurve 2D graph with the **frequency domain decomposition** in the upper graph and the input and output time domain data in the lower graph.



The graph's toolbar has a button that toggles on and off and changes the proportions of the two graphs in the dialog. The default has the Fourier graph using the upper half of the region and the output graph using the lower half.

Algorithm

This procedure uses the **Best Exact n FFT** procedure to create the frequency spectrum in the upper graph. The **Best Exact n FFT** procedure is also used for the FFT inverse that reconstructs the time domain data from the filtered frequency domain representation. The spectrum is always displayed in a **normalized decibel format**.



If the data set size is 512 or lower (257 or lower points in the frequency spectrum), a bar format is automatically used. Otherwise, the points in the spectrum will be rendered directly with lines connecting the points. You can override either of these point formats using the **Modify Point Format** option and their respective states will be saved across sessions.

In this procedure, the main task is to set a threshold, either in frequency or spectral magnitude, whereby the noise that is present in the signal can be removed. When a **Frequency** threshold is selected, all frequency domain information beyond this specified frequency is zeroed in order to create the output data. It is quite common for a signal to exist at low frequencies, whereas white noise exists across all frequencies. This low-pass filtration preserves the signal and discards the noise within the higher frequencies. The noise that exists within the retained signal frequencies is not filtered.

Unlike time domain filters, a Fourier domain lowpass filter can achieve an absolute truncation at the specified frequency. Whereas discontinuities in time domain data introduce real problems with Fourier analysis, this "cliff" or "brickwall" discontinuity that is produced in the Fourier spectrum of the output data is not usually a concern. In fact, it is usually the ideal in terms of noise reduction and smoothing. If the output data are subsequently transformed into the Fourier domain using an exact-n FFT, it will look exactly like the FFT for the active or retained channels in the upper graph. The spectral magnitudes for the frequencies that had been zeroed will exist at approximately -300dB, the machine precision limit, in a normalized dB plot.

If a **dB** threshold is selected, the frequency channels are zeroed based on a specified signal threshold. If the signal of interest can be assumed to exist primarily in the highest dB Fourier channels, as is often the case, it is possible that a much higher percentage of the overall channels can be zeroed. This is the only way to threshold a signal with high frequency components within the Fourier domain.

If possible, it is usually a good idea to retain the adjacent sidelobes of each spectral component.

Unlike the frequency thresholding, dB thresholding can result in multiple "cliff" or "brickwall" discontinuity zones in the Fourier spectrum of the output data. This is especially true when multiple components are present. There may be any number of zones at approximately -300dB in a normalized dB spectrum. Again, this is generally the ideal for removing noise.

If you really need to produce soft transitions between the retained and discarded frequencies, you can use the [Fourier Filtering](#) procedure with a data tapering window to implement this same type of frequency or dB thresholding. When this is done, a subsequent transform of the filtered data into the Fourier domain using an exact-n non-windowed FFT will produce the transitions typical of time-domain filter algorithms.

The dB thresholding, in one simple step, accomplishes what could require a number of bandpass filters in the time domain and it does so with brickwall cutoffs. In this basic procedure, the dB threshold is fixed for all components. To filter using a different dB threshold for each component, the [Fourier Filtering](#) option must be used.

Thresholding

The Frequency and dB threshold values can be entered numerically or graphically. To enter a frequency graphically left click the mouse on the upper graph at the frequency desired (the dB position is unimportant). To enter a dB value graphically, left click the mouse in the upper graph at the desired dB (the frequency is unimportant). In this option, it is not possible to graphically toggle individual frequency channels on and off.



The [Reset Previous](#) button restores the threshold value last used in this procedure.

Removal of Linear Trend

Check the [Remove Trend](#) box to remove a two point (initial and final data value) linear trend from the data stream. The data trend is automatically restored in the reconstruction.

Data Tapering Window

"Raised" [data taper windows](#) (those that are non-zero at the bounds) are an effective way to reduce spectral leakage and achieve a more effective noise removal. They should be used with caution since a taper discards much of the information near the bounds of the data.

The following [data taper windows](#) are offered:

- None
- cs2 Hamming
- Chebyshev 2
- Chebyshev 2.5
- cs3 BHarris3
- cs3 Blckmn Exact
- cs3 Nutall min
- Chebyshev 3
- Chebyshev 3.5

The full selection of data windows is offered in the [Fourier Filtering](#) option.

Estimated Noise Reduction

TableCurve 2D offers a robust noise estimation procedure that may be of some value for low-frequency signals. A cubic polynomial interpolation is made for each point using the two points to the left and the two to the right (excluding the current point). The difference between the interpolated and data values is

used to generate a measure of the white noise present in the data. This assumes that the data can be locally characterized by a smooth cubic interpolant. Also, the signal component(s) should exist only in the lower quarter of the Nyquist range. If a high frequency signal component is present, these estimates of noise will be invalid.

The **In** value reports the estimated white noise in the incoming data, the **Out** value the estimated white noise for the filtered signal. The percent is given as the amount of estimated noise remaining after filtration.

Power Reduction

In this section the **In** value reports the TISA (time-integral squared amplitude) power in the incoming data. The **Out** value is the TISA power for the filtered signal. The percent is given as the amount of power remaining after filtration. For most S/N ratios, the reduction in power should be minimal. These values can alert you when signal components are being discarded.

Correlation Coefficient

The **r-squared correlation coefficient** should also remain high. An r^2 of 1 is a perfect correlation while a value of 0 means the filtered and unfiltered signals are uncorrelated. Low r^2 values are also indicative of signal components being lost in the thresholding.

List



The **List Data** option lists the index, time, and output signal in a three column table. The listing uses the **TableCurve 2D text viewer** facility.

Copy



The **Copy Data to Clipboard** option copies the time and output signal values to the clipboard. Formats include full precision binary (for spreadsheets such as Excel) and ASCII (for pasting into text editors).

Save



The **Save Data to Disk** option writes the time and output values to a supported file format. These formats include ASCII, Excel 97, Excel 95, Lotus WK3, Lotus WK1, SPSS, or Systat.

Production Facility



The **TableCurve 2D Automation facility** allows unattended processing of large numbers of data sets. The data sets can be consolidated in an Excel file or acquired using a DLL. The graphs and reports can be exported to an MS Word/RTF file, while the processed data can be exported to an Excel 95 or Excel 97/2000 file.



The **Reset** button restores the data to its state when first entering the procedure. If an Automation Session is in progress, the **Reset** button can be used to terminate the automated processing.

Residuals



The **Residuals** button is used to graphically display the difference between the original and filtered data.

Numeric Summary



The **Numeric Summary** offers a full FFT report. The report optionally includes a listing of the interpolated spectral peaks, a frequency analysis, and a linear sinusoidal least-squares fit summary.

FFT Summary

 The **List Frequency Domain Data** option offers an extended FFT data summary. The listing uses the **TableCurve 2D text viewer** facility. The FFT channel number, frequency, and magnitude are always listed. The **Format** menu offers the optional selection of the following:

- Add Real, Imag
- Add Amplitude
- Add Wavelength
- Add Phase (Sine-based)
- Add dB
- or Add dB Normalized
- Add Power Spectral Density, Sum Squared Amplitude
- or Add Power Spectral Density, Mean Squared Amplitude
- or Add Power Spectral Density, Time-Integral Squared Amplitude

In the following table, **Re** is the real component of the FFT at a given frequency, **Im** is the imaginary component, **n** is the data set size, and **dx** is the sampling interval:

- Magnitude, $\sqrt{\text{Re}*\text{Re}+\text{Im}*\text{Im}}$
- Phase, sine-based, $\text{Pi}/2+\text{atan}(\text{Im}/\text{Re})$
- Amplitude, $2.0*\sqrt{\text{Re}*\text{Re}+\text{Im}*\text{Im}}/n$
- dB, decibels, $10.0*\log_{10}(\text{Re}*\text{Re}+\text{Im}*\text{Im})$
- dB Norm, decibels, normalized to 0 for frequency channel with maximum power
- PSD SumSq, Power as Sum Squared Amplitude, $2.0*(\text{Re}*\text{Re}+\text{Im}*\text{Im})/n$
- PSD MeanSq, Power as Mean Squared Amplitude, $2.0*(\text{Re}*\text{Re}+\text{Im}*\text{Im})/n/n$
- PSD TimeInt, Power as Time-Integral Squared Amplitude, $2.0*dx*(\text{Re}*\text{Re}+\text{Im}*\text{Im})/n$

In an amplitude plot, you see the actual amplitude of sine components. In a normalized decibel plot, the highest peak is at 0dB, a peak at -3dB would have half the power, and a peak at -6dB would have half the amplitude. The PSD TISA (time-integral squared amplitude power) is the actual integral under the curve defined by the square of the raw data.

The amplitude and phase of each component in the FFT is derived from sine-based conversion. Each of the components in the FFT can be reconstructed using:

$$Y = \text{Amplitude} * \sin(2 * \text{PI} * \text{Frequency} * X + \text{Phase})$$

MS Word/RTF Export

 The **MS Word/RTF File Export** option is used to save the current graphs and reports to either an MS Word file or a portable RTF (Rich Text Format) file. The graphs are inserted into the file as Windows metafiles.

Automatic Update

The List, Residuals, Numeric Summary, and Fourier Summary windows are automatically updated when any change is made to the algorithm's settings.

Updating the Data Table

 When exiting this procedure with the **OK** button, an option will be presented to update TableCurve

2D's main data table with the denoised data.

If **Background Thread Fitting** is active, the fitting will be initiated as soon as the filtering is accepted and the data are rescanned.

Eigendecomposition Denoising

The **Eigendecomposition Denoising** option in the Filter menu or the main toolbar is a specialized **Eigendecomposition** filtration procedure that sets a signal strength threshold for retaining all signal-bearing eigenmodes and filtering out all noise-containing ones. This option is used exclusively to remove noise. In general, eigenfiltration is the most effective method for denoising.

Uniformly spaced data are needed so that no additional noise is introduced. This procedure does not automatically generate uniform data for the Eigendecomposition. If needed, the **Spline Estimation** or **Local Regression Spline Estimation** option can be used to create an interpolated data set with constant X-spacing.

The option presents a dual **TableCurve 2D graph** with the eigenvalues in the upper graph and the input and output time domain data in the lower graph.



The graph's toolbar has a button that can toggle on and off and change the proportions of the two graphs in the dialog. The default has the eigenvalue graph using the upper third of the region, and the output graph using the lower two-thirds.

Algorithm

The **eigendecomposition** algorithms vary only by how a lagged covariance matrix is constructed:

- **CovM Fwd** - the covariance matrix equivalent of processing a forward prediction data matrix
- **CovM FB** - the covariance matrix equivalent of processing a forward-backward prediction data matrix
- **CovM Tpltz** - a Toeplitz symmetric covariance matrix constructed using the Burg procedure

In general, the differences between the methods will be small. Because it imposes a matrix structure, the **CovM Tpltz** algorithm is typically the least effective in mapping data trends. If the **AR algorithm** properties apply, the **CovM FB** algorithm should be the most effective of the procedures.

Full data matrix procedures are not needed since no parametric model is computed. The decomposition of a forward or forward-backward data matrix produces the same eigenvalues and eigenvectors as the equivalent covariance matrix, and requires more processing time with large data sets. Similarly, no backward prediction data/covariance matrix algorithm is needed since it would produce the same decomposition as the forward data/covariance matrix.

In this procedure, the main task is to set a sorted eigenvalue threshold whereby the noise that is present in the signal can be removed. All of the eigenmodes are used in the reconstruction. The principal components of the signal will be captured in the initial eigenmodes. The noise is broken into low power elements that generally form a sloping floor beyond the signal eigenmodes in the eigenvalue plot.

Order

Establishing the **eigendecomposition** order for harmonic signals in the absence of noise is a simple matter. A model order of two is needed to fully describe one oscillation. Similarly an order of four is needed to fully model two oscillatory components. For noise-free data, the minimum order needed is twice the number of signal components introducing oscillations in the data. These oscillatory components can be harmonic, such as undamped or damped sinusoids, sawtooth, and others, or they can be anharmonic: oscillations where the model is not readily apparent.

Since there is usually some level of noise present in the data, a higher order model is needed to also account for the oscillations introduced by noise. To achieve a reasonable signal-noise separation within an eigendecomposition, it is necessary to use a high enough order so that the primary eigenvectors span only signal space. In general, the order is not particularly critical so long as it is sufficient to produce an

effective partitioning of the signal and noise. What is crucial is the accurate specification of the **signal space threshold**.

Signal Subspace Selection

The number of principal eigenmodes comprising signal components can be specified numerically if you know with a certainty the number of spectral components present in the data. Again, you must enter a value that is twice the number of components. If three spectral components are known to exist, the signal subspace must be set to six. Even when the spectral component count is known, you should use the graphical inspection of eigenvalues to insure that a high enough order is being used to achieve the desired signal-noise separation.

When there is sufficient **signal-noise separation** in the eigenmodes, the eigenvalue plot reveals one or more sharp transitions between the signal subspace and the noise subspace floor. The last eigenmode before the long sloping noise floor represents the last element of signal space. Assuming a high-enough eigendecomposition order is used, this signal-noise separation does not become difficult until the noise level approaches that of the signal. At this point, the sharp characteristic transition may disappear.

To select the **signal space** graphically, left click on the last of the eigenvalues judged to represent signal space. All of the eigenvalues used in the reconstruction will be circled.

Estimated Noise Reduction

TableCurve 2D offers a robust noise estimation procedure that may be of some value for low-frequency signals. A cubic polynomial interpolation is made for each point using the two points to the left and the two to the right (excluding the current point). The difference between the interpolated and data values is used to generate a measure of the white noise present in the data. This assumes that the data can be locally characterized by a smooth cubic interpolant. Also, the signal component(s) should exist only in the lower quarter of the Nyquist range. If a high frequency signal component is present, these estimates of noise will be invalid.

The **In** value reports the estimated white noise in the incoming data, the **Out** value the estimated white noise for the filtered signal. The percent is given as the amount of estimated noise remaining after filtration.

Power Reduction

In this section the **In** value reports the TISA power in the incoming data. The **Out** value is the TISA power for the filtered signal. The percent is given as the amount of power remaining after filtration. For most S/N ratios, the reduction in power should be minimal. These value can alert you when signal components are being discarded.

Correlation Coefficient

The **r-squared correlation coefficient** should also remain high. An r^2 of 1 is a perfect correlation while a value of 0 means the filtered and unfiltered signals are uncorrelated. Low r^2 values are also indicative of signal components being lost in the thresholding.

List



The **List Data** option lists the index, time, and output signal in a three column table. The listing uses the **TableCurve 2D text viewer** facility.

Copy



The **Copy Data to Clipboard** option copies the time and output signal values to the clipboard. Formats include full precision binary (for spreadsheets such as Excel) and ASCII (for pasting into text editors).

Save



The **Save Data to Disk** option writes the time and output values to a supported file format. These formats include ASCII, Excel 97/2000, Excel 95, Lotus WK3, Lotus WK1, SPSS, or Systat.

Production Facility



The TableCurve 2D Automation facility allows unattended processing of large numbers of data sets. The data sets can be consolidated in an Excel file or acquired using a DLL. The graphs and reports can be exported to an MS Word/RTF file, while the processed data can be exported to an Excel 95 or Excel 97/2000 file.



The **Reset** button restores the data to its state when first entering the procedure. Note that if you implement sequential local procedures, all of the revisions are discarded upon reset. If an Automation Session is in progress, the **Reset** button can be used to terminate the automated processing.

Residuals



The **Residuals** button is used to graphically display the difference between the original and filtered data.

Numeric Summary



The **Numeric Summary** offers an eigendecomposition report. The report includes a table of eigenvalues, normalized eigenvalues, and partial r^2 correlation coefficients.

MS Word/RTF Export



The **MS Word/RTF File Export** option is used to save the current graphs and reports to either an MS Word file or a portable RTF (Rich Text Format) file. The graphs are inserted into the file as Windows metafiles.

Automatic Update

The List, Residuals, and Numeric Summary windows are automatically updated when any change is made to the algorithm's settings.

Updating the Data Table



When exiting this procedure with the **OK** button, an option will be presented to update TableCurve 2D's main data table with the filtered data.

If **Background Thread Fitting** is active, the fitting will be initiated as soon as the filtering is accepted and the data are rescanned.

Fourier Filtering

The [Fourier Filtering](#) option in the Filter menu or the main toolbar is an extensive Fourier domain filtering and component isolation procedure. In this option, it is possible to set lower and upper frequency thresholds as well as lower and upper spectrum thresholds, either numerically or graphically. Individual Fourier channels can also be toggled on and off graphically. This option also offers reconstruction of windowed FFTs for isolating very low power components.

Uniformly spaced data are needed so that no additional noise is introduced. This procedure does not automatically generate uniform data for the Fourier decomposition. If needed, the [Spline Estimation](#) or [Local Regression Spline Estimation](#) option can be used to create an interpolated data set with constant X-spacing.

This procedure uses an inverse FFT for the reconstruction. The filtered or reconstructed signal will thus contain the same number of data elements as the original or unfiltered signal. To generate any size reconstruction or to reconstruct a specific range of data, a similar (but non-windowed) filtering and reconstruction can be found in the [Fourier Estimation](#) option. There the reconstruction is made directly from the amplitude, frequency, and phase of the continuous sine components rather than by an inverse FFT.

This option presents a dual [TableCurve 2D graph](#) with the [frequency domain decomposition](#) in the upper graph and the input (unfiltered) and output (filtered) time domain data in the lower graph.



The graph's toolbar has a button that can toggle on and off and change the proportions of the two graphs in the dialog. The default has the Fourier graph using the upper half of the region, and the output graph using the lower half.

Window

The use of a data window in a Fourier domain filter procedure is not a normal practice. There are two instances, however, where a data tapering window is beneficial. The most common use of data windows is to minimize spectral leakage so that very low power components can be isolated and reconstructed. When no data tapering window is used, low energy components are lost in the background. The use of a data tapering window enhances the dynamic range of the FFT so that even components of very low power can be clearly distinguished above a noise background.

The other reason is to produce softer filter transitions. When some frequencies are zeroed within an exact n non-windowed FFT, and the reconstructed data are again processed by an exact n non-windowed FFT, there will be true brickwall transitions at the zeroed frequencies. While this is generally desirable, and even the ideal, the transition observed in the subsequent exact n non-windowed FFT can be softened by using a windowed FFT as the basis for the initial processing (of course, the same brickwall occurs if the data window is used for both FFTs).

TableCurve 2D offers a broad selection of [data tapering windows](#) to [minimize spectral leakage](#). The [adj](#) field is used to set the spectral width, and thus the dynamic range, of adjustable windows. This field will be disabled for fixed windows.



An appropriate window and adjustable width can be determined by using the [Explore Data Tapering Windows](#) option.

In TableCurve 2D, a windowed FFT is normalized so that its power equals the power of the input data, regardless of the data window used.

When a window is used in this option, an [edge%](#) value must be specified. This is a tolerance below which

no reconstruction occurs (data tapered to zero at the bounds cannot be reconstructed). The reconstructed data will be shortened in length by the removal of points near the bounds that occurs as a consequence of this edge tolerance. Two windows of particular merit in this option are the **Hamming** and **Blackman** windows since they do not fully decay to zero at the bounds of the taper and can usually reconstruct the full data range at the default 1% edge tolerance.

Transform

TableCurve 2D offers the following FFT algorithms:

- **FFT Radix 2**
- **Prime Factor**
- **Mixed Radix**
- **Chirp-Z**
- **Best Exact N**

The **Best Exact N** composite algorithm is the default. If the data size is a power of 2, the **FFT Radix 2** algorithm is used. If not and the size is included in the prime-factor set, then the **Prime Factor** procedure is used. Otherwise, the **Mixed Radix** algorithm is used if the largest prime ≤ 509 and the **Chirp-Z** is used if the largest prime ≥ 521 . This produces the fastest possible exact n FFT.

Data reconstruction is via an inverse FFT. The same algorithm is used for both the forward and reverse transforms.

Nmin

The initial value will be the data size. To zero pad, enter any value greater than the data size. You may also select from a power of 2 sequence in the drop down box. Note that the actual size of the FFT may be greater than this value if the **FFT Radix 2** or **Prime Factor** algorithm is used.

Increasing the zero padding will increase the number of frequency channels, which may improve the accuracy of isolating components or separating signal and noise for reconstructions.

Plot

The frequency domain information can be plotted in a variety of formats. In the following table, Re is the real component of the FFT at a given frequency, Im is the imaginary component, n is the data set size, dx is the sampling interval, and var is the variance of the data series.

- **Magnitude**, $\sqrt{\text{Re}^2 + \text{Im}^2}$
- **Amplitude**, $2.0 * \sqrt{\text{Re}^2 + \text{Im}^2} / n$
- **Phase**, sine-based, $\pi/2 + \arctan(\text{Im}/\text{Re})$
- **Variance**, Power normalized by variance, $(\text{Re}^2 + \text{Im}^2) / n / \text{var}$
- **dB**, decibels, $10.0 * \log_{10}(\text{Re}^2 + \text{Im}^2)$
- **dB Norm**, decibels, normalized to 0 for frequency channel with maximum power
- **PSD SumSq**, Power as Sum Squared Amplitude, $2.0 * (\text{Re}^2 + \text{Im}^2) / n$
- **PSD MeanSq**, Power as Mean Squared Amplitude, $2.0 * (\text{Re}^2 + \text{Im}^2) / n / n$
- **PSD TimeInt**, Power as Time-Integral Squared Amplitude, $2.0 * dx * (\text{Re}^2 + \text{Im}^2) / n$

In an amplitude plot, you see the actual amplitude of sine components. In a normalized decibel plot, the highest peak is at 0dB, a peak at -3dB would have half the power, and a peak at -6dB would have half the amplitude. The PSD TISA (time-integral squared amplitude power) is the actual integral under the curve defined by the square of the raw data.



If the data set size is 512 or lower (257 or lower pts in the frequency spectrum), a bar format is

automatically used. Otherwise, the points in the spectrum will be rendered directly with lines connecting the points. You can override either of these point formats using the Modify Point Format option and their respective states will be saved across sessions.

Frequency Thresholds

There are several ways to set the thresholds for Fourier filtering. The first is to define a range of frequency space. This is primarily used to isolate and reconstruct signal components. The [Freq1](#) and [Freq2](#) fields set limits in frequency. The frequency boundaries of this region can either be included ([Incl](#)) or excluded ([Excl](#)). If a frequency range is included, this means that only the frequencies that fall within this range are used in the data reconstruction. If the region is excluded, only those frequencies that fall outside the region are used for the reconstruction.

The frequencies can be entered numerically. If both fields for frequency are left blank, the full range is assumed for the [Incl](#) option, and no range is specified for the [Excl](#) option. If only a lower limit is entered, the variable range is assumed to be from this limit and greater for the [Incl](#) option, and all values below this limit if the [Excl](#) option is set. If only an upper limit is entered, the variable range is assumed to be from this limit and lower for [Incl](#), and all values above this limit for [Excl](#).



It is generally easier to set a frequency range graphically. When the [X Sectioning](#) mode is selected, the mouse is used in the upper graph to select the frequency range of interest. Frequencies are toggled off by holding the left mouse button down and sliding right and toggled back on by moving left. Individual frequencies can be toggled on and off by placing the cursor at the point (or top of a bar) and left clicking the mouse.

Spectral Thresholds

Another way to set the thresholds for Fourier filtering involves defining spectral thresholds. This is primarily used to separate signal components by power. It can also be used for noise removal, replicating the functionality in the [Fourier Denoising](#) option, or for signal removal, generating a signal-free background for subsequent analysis.

The [Spec1](#) and [Spec2](#) fields set the spectral limits. These can only be entered numerically. If only [Spec1](#) is set, all values at or above this threshold will be used in the reconstruction if [Incl](#) is set, and all below this threshold will be used if [Excl](#) is checked. If only [Spec2](#) is set, all values at this threshold and below will be used in the reconstruction if [Incl](#) is set, and all above this threshold will be used if [Excl](#) is set. For this option, the normalized dB values must use the minus sign.

Frequency-Spectral Thresholds

The thresholds for Fourier filtering can also consist of a frequency-spectrum region. This allows signal components to be isolated by both frequency and spectral power. This is the most effective way to isolate and reconstruct a single signal component when some number of components are present.

The [Freq1](#) and [Freq2](#) fields set limits in frequency while the [Spec1](#) and [Spec2](#) fields set the spectral limits. The values can be entered numerically. If only a lower limit field is set, all values at or above this threshold will be used in the reconstruction if [Incl](#) is set, and all below this threshold will be used if [Excl](#) is checked. If only the upper limit is set, all values at this threshold and below will be used in the reconstruction if [Incl](#) is set, and all above this threshold will be used if [Excl](#) is set. Again, the normalized dB values must use the minus sign.



It is generally easier to set a frequency-spectrum region graphically. When the [XY Sectioning](#) mode is selected, the mouse is used in the upper graph to enclose the frequency-spectral region of interest. To include the region that is to be graphically defined, click and hold down the left mouse button at one corner of the frequency-spectral rectangle in the upper graph and drag the rectangle so that it encloses the region desired. This will automatically enter both the frequency and spectral limits and set the process

state to include the FFT channels located within the region.

To exclude the region that is to be graphically defined, the same graphical procedure is followed except that the right rather than the left mouse button is used. In this case, the process state will be set to exclude the FFT channels within the region. This is useful for reconstructing data with a single component, such as line noise, removed.

Note that it is possible to reconstruct only one frequency-spectral region at a time.



The **Clear** button is used to clear all of the threshold fields.



The **Apply** button can be used to immediately update the processing rather than wait for the automatic update.

Popup Information



The **Toggle Popup Information Window** option is used to report estimated noise and power reductions as well as a correlation coefficient between the filtered and unfiltered data.

TableCurve 2D offers a robust noise estimation procedure that may be of some value for low-frequency signals. A cubic polynomial interpolation is made for each point using the two points to the left and the two to the right (excluding the current point). The difference between the interpolated and signal values is used to generate a measure of the white noise present in the signal. This assumes that the signal can be locally characterized by a smooth cubic interpolant. Also, the signal component(s) should exist only in the lower quarter of the Nyquist range. If a high frequency signal component is present, these estimates of noise will be invalid. The **Noise In** value reports the estimated white noise in the incoming data, the **Noise Out** value reports the estimated white noise for the filtered signal. The **Noise %** is given as the amount of estimated noise remaining after filtration.

The **TISA In** value reports the TISA power in the incoming data. The **TISA Out** value is the TISA power for the filtered signal. The **TISA %** is given as the amount of power remaining after filtration.

The **r-squared correlation coefficient** is also reported. An r^2 of 1 is a perfect correlation while a value of 0 means the filtered and unfiltered signals are completely uncorrelated.

List



The **List Data** option lists the index, time, and output signal in a three column table. The listing uses the **TableCurve 2D text viewer** facility.

Copy



The **Copy Data to Clipboard** option copies the time and output signal values to the clipboard. Formats include full precision binary (for spreadsheets such as Excel) and ASCII (for pasting into text editors).

Save



The **Save Data to Disk** option writes the time and output values to a supported file format. These formats include ASCII, Excel 97/2000, Excel 95, Lotus WK3, Lotus WK1, SPSS, or Systat.

Production Facility



The **TableCurve 2D Automation** facility allows unattended processing of large numbers of data sets. The data sets can be consolidated in an Excel file or acquired using a DLL. The graphs and reports can be exported to an MS Word/RTF file, while the processed data can be exported to an Excel 95 or Excel

97/2000 file.



The **Reset** button restores the data to its state when first entering the procedure. If an **Automation Session** is in progress, the **Reset** button can be used to terminate the automated processing.

Residuals



The **Residuals** button is used to graphically display the difference between the original and filtered data.

Numeric Summary



The **Numeric Summary** offers a full FFT report. The report optionally includes a listing of the interpolated spectral peaks, a frequency analysis, and a linear sinusoidal least-squares fit summary.

FFT Summary



The **List Frequency Domain Data** option offers an extended FFT data summary. The listing uses the **TableCurve 2D text viewer** facility. The FFT channel number, frequency, and magnitude are always listed. The **Format** menu offers the optional selection of the following:

- Add Real, Imag
- Add Amplitude
- Add Wavelength
- Add Phase (Sine-based)
- Add dB
- or Add dB Normalized
- Add Power Spectral Density, Sum Squared Amplitude
- or Add Power Spectral Density, Mean Squared Amplitude
- or Add Power Spectral Density, Time-Integral Squared Amplitude

In the following table, **Re** is the real component of the FFT at a given frequency, **Im** is the imaginary component, **n** is the data set size, and **dx** is the sampling interval:

- **Magnitude**, $\sqrt{\text{Re} \cdot \text{Re} + \text{Im} \cdot \text{Im}}$
- **Phase**, sine-based, $\text{Pi}/2 + \text{atan}(\text{Im}/\text{Re})$
- **Amplitude**, $2.0 \cdot \sqrt{\text{Re} \cdot \text{Re} + \text{Im} \cdot \text{Im}}/n$
- **dB**, decibels, $10.0 \cdot \log_{10}(\text{Re} \cdot \text{Re} + \text{Im} \cdot \text{Im})$
- **dB Norm**, decibels, normalized to 0 for frequency channel with maximum power
- **PSD SumSq**, Power as Sum Squared Amplitude, $2.0 \cdot (\text{Re} \cdot \text{Re} + \text{Im} \cdot \text{Im})/n$
- **PSD MeanSq**, Power as Mean Squared Amplitude, $2.0 \cdot (\text{Re} \cdot \text{Re} + \text{Im} \cdot \text{Im})/n/n$
- **PSD TimeInt**, Power as Time-Integral Squared Amplitude, $2.0 \cdot dx \cdot (\text{Re} \cdot \text{Re} + \text{Im} \cdot \text{Im})/n$

In an amplitude plot, you see the actual amplitude of sine components. In a normalized decibel plot, the highest peak is at 0dB, a peak at -3dB would have half the power, and a peak at -6dB would have half the amplitude. The PSD TISA (time-integral squared amplitude power) is the actual integral under the curve defined by the square of the raw data.

The amplitude and phase of each component in the FFT is derived from sine-based conversion. Each of the components in the FFT can be reconstructed using:

$$Y = \text{Amplitude} \cdot \sin(2 \cdot \text{PI} \cdot \text{Frequency} \cdot X + \text{Phase})$$

MS Word/RTF Export



The **MS Word/RTF File Export** option is used to save the current graphs and reports to either an MS Word file or a portable RTF (Rich Text Format) file. The graphs are inserted into the file as Windows metafiles.

Automatic Update

The List, Residuals, Numeric Summary, and Fourier Summary windows are automatically updated when any change is made to the algorithm's settings.

Updating the Data Table



When exiting this procedure with the **OK** button, an option will be presented to update TableCurve 2D's main data table with the filtered data.

If **Background Thread Fitting** is active, the fitting will be initiated as soon as the filtering is accepted and the data is rescanned.



Eigendecomposition Filtering

The **Eigendecomposition Filtering** option in the Filter menu or the main toolbar offers full eigenmode filtering and reconstruction. **Eigendecomposition** partitions by signal strength rather than by frequency. Using this option, it is possible to isolate individual oscillatory components in signals. It is also possible to **isolate the signal and noise portions** of a data series.

Uniformly spaced data are needed so that no additional noise is introduced. This procedure does not automatically generate uniform data for the Eigendecomposition. If needed, the **Spline Estimation** or **Local Regression Spline Estimation** option can be used to create an interpolated data set with constant X-spacing.

This option presents a dual **TableCurve 2D graph** with the eigenvalues in the upper graph and a variety of visualization options in the lower graph. In addition to the reconstructed data, the lower graph can optionally consist of the eigenvectors, the principal components, the data components, FFTs of the data components, or the FFT of the data.



The graph's toolbar has a button that can toggle on and off and change the proportions of the two graphs in the dialog. The default has the eigenvalue graph using the upper third of the region, and the output graph using the lower two-thirds.

Graphical Selection of Eigenmodes



The eigenmode selection in the upper graph is fully graphical. To select eigenmodes, the **Selection Mode** must be active. All eigenvalues boxed using the left mouse button are included in the reconstruction while those outside the enclosed region are excluded. If the right mouse button is used to enclose a set of eigenvalues, these are excluded from the reconstruction and all others are included. Individual eigenmodes can be turned on and off by left or right clicking on the point. To select a single eigenmode for reconstruction, it is necessary that you box just that eigenmode using the left mouse button. To exclude just a single eigenmode, you must similarly box the desired mode using the right mouse button.

Two eigenmodes are required to capture an oscillation. For this reason eigenmodes often appear in pairs and should be selected and processed in pairs. Note also that an **eigendecomposition** is non-parametric. A non-sinusoidal anharmonic oscillation is captured as easily as a sinusoidal harmonic.

Eigenvalue Data Labels



The **Toggle Data Labels** option displays an **r^2 correlation coefficient** for each of the eigenmodes used in the reconstruction. These partial r^2 values will generally follow the ordering of the eigenvalues. These r^2 values measure the extent to which each eigenmode captures the variance in the overall data. A perfect r^2 is 1.0. This will not be observed with oscillatory data since two eigenmodes are needed to capture a single oscillatory component. An r^2 of 0 indicates no correlation at all between the data set and the data from the reconstructed eigenmode. Eigenmodes containing noise will usually have partial r^2 values very near zero. Note that the partial r^2 values derive from the reconstruction of individual components.

Algorithm

The **eigendecomposition** algorithms vary only by how a lagged covariance matrix is constructed:

- **CovM Fwd** - the covariance matrix equivalent of processing a forward prediction data matrix
- **CovM FB** - the covariance matrix equivalent of processing a forward-backward prediction data matrix
- **CovM Tpltz** - a Toeplitz symmetric covariance matrix constructed using the Burg procedure

In general, the differences between the methods will be small. Because it imposes a matrix structure, the

CovM Tpltz algorithm is typically the least effective in mapping data trends. If the **AR algorithm** properties apply, the **CovM FB** algorithm should be the most effective of the procedures.

Full data matrix procedures are not needed because no parametric model is computed. The decomposition of a forward or forward-backward data matrix produces the same eigenvalues and eigenvectors as the equivalent covariance matrix, and requires more processing time with large data sets. Similarly, no backward prediction data/covariance matrix algorithm is needed since it would produce the same decomposition as the forward data/covariance matrix.

In this procedure, the main task is to isolate and reconstruct signal components of interest by selecting one or more eigenmodes. The principal components of the signal will be captured in the initial eigenmodes. The noise is broken into low power elements that generally form a sloping floor beyond the signal eigenmodes in the eigenvalue plot.

Order

Establishing the **eigendecomposition** order for harmonic signals in the absence of noise is a simple matter. A model order of two is needed to fully describe one oscillation. Similarly an order of four is needed to fully model two oscillatory components. For noise-free data, the minimum order needed is twice the number of signal components introducing oscillations in the data. These oscillatory components can be harmonic, such as undamped or damped sinusoids, sawtooth, and others, or they can be anharmonic, oscillations where the model is not readily apparent.

Since there is usually some level of noise present in the data, a higher order model is needed to also account for the oscillations introduced by noise. To achieve a reasonable signal-noise separation within an eigendecomposition, it is necessary to use a high enough order so that the primary eigenvectors span only signal space. Further, when isolating components, the partitioning of component signals into different eigenmodes is usually enhanced by higher orders.

When this procedure is invoked, all of the eigenmodes in the decomposition are automatically selected. This produces an exact reconstruction of the input data.

Normalize

The **Normalize (%)** item plots the eigenvalues on a percent scale. When this normalization is used, all of the eigenvalues sum to 100%. This option makes it easy to see the relative contribution, by eigenvalue, of each component.

Eigenvectors

When the **Eigenvectors** option is selected, each of the eigenvectors used in the reconstruction is plotted. Each eigenvector plotted will be of the length of the embedding dimension (model order). Sinusoidal harmonics will generally appear in pairs, one offset in index from the other.

Principal Components

The **Princ. Comp.** option plots the temporal principal components used in the reconstruction. The temporal components consist of the projection of the eigenvectors onto the data series. Each temporal component plotted will be of the length of the data series minus the model order plus one. Again, sinusoidal harmonics will generally appear in pairs, one offset in index from the other. Unlike the eigenvectors, the magnitude of the principal components will be related to signal strength.

Data Components

The **Data Comp.** option plots the reconstructed data components. This is normally the starting point for component inspection, as compared to the eigenvectors or principal components. Each data component plotted will be of the length of the data series and each element will correspond in time with the original data. Phase is also fully restored. Separated sinusoidal harmonics will still appear in pairs, although the

data plots will now closely overlap one another. The offset observed in the eigenvectors or principal components representing an oscillatory pair will not be present in the reconstructed data components.

Data

The [Data](#) option plots the overall reconstruction of data using all selected eigenmodes. There is but a single plot.



The [Toggle Display of Reference Data](#) button in the [TableCurve 2D graph's toolbar](#) is used to toggle the original data on and off.

FFT Components

The [FFT Comp.](#) option is a convenience for inspecting the Fourier spectra of the individual reconstructed data components. For these frequency spectra, a windowed FFT is offered. A Kaiser-Bessel window is used, its variable width automatically determined as a function of data count. The width will be 1.5 for $n \leq 256$, and increases to 4.0 for $n \geq 2048$. Since the windowing takes care of edge effects, a minimum 1024 point FFT is used to give a good peak resolution with short data records. For data sets beyond 1024 points, an exact n FFT is used. There will thus be 513 points in each spectrum for all data sets 1024 points in size and smaller. Larger sets will result in $n/2+1$ frequencies. A decibel (dB) scale is used so that individual components can be compared for power.

FFT

The [FFT](#) option is used for inspecting the Fourier spectrum of the overall data reconstruction. The same Kaiser-Bessel windowing, 1024 minimum size FFT, and dB scaling is used.



The [Toggle Display of Reference Data](#) button in the [TableCurve 2D graph's toolbar](#) is used to toggle the FFT of the original data on and off.

Component Labels



When components are plotted, eigenmode numbers can be plotted as data labels. These are automatically positioned at the peaks in the data. The [Toggle Reference Labels](#) button in the [TableCurve 2D graph's toolbar](#) is used to toggle these labels on and off.

Popup Information



The [Toggle Popup Information Window](#) option is used to report estimated noise and power reductions as well as a correlation coefficient between the filtered and unfiltered data.

TableCurve 2D offers a robust noise estimation procedure that may be of some value for low-frequency signals. A cubic polynomial interpolation is made for each point using the two points to the left and the two to the right (excluding the current point). The difference between the interpolated and data values is used to generate a measure of the white noise present in the data. This assumes that the signal can be locally characterized by a smooth cubic interpolant. Also, the signal component(s) should exist only in the lower quarter of the Nyquist range. If a high frequency signal component is present, these estimates of noise will be invalid.

The [Noise In](#) value reports the estimated white noise in the incoming data, the [Noise Out](#) value the estimated white noise for the filtered signal. The [Noise %](#) is given as the amount of estimated noise remaining after filtration.

The [TISA In](#) value reports the TISA power in the incoming data. The [TISA Out](#) value is the TISA power for the filtered signal. The [TISA %](#) is given as the amount of power remaining after filtration.

The **r-squared correlation coefficient** is also reported. An r^2 of 1 is a perfect correlation while a value of 0 means the filtered and unfiltered signals are completely uncorrelated.

List



The **List Data** option lists the index, x-values, and the output currently plotted in a table that includes up to 12 components. The listing uses the **TableCurve 2D text viewer** facility. Note that all information will not be present if more than 12 eigenmodes are reconstructed.

Copy



The **Copy Data to Clipboard** option copies the x values and the output currently plotted to the clipboard. Formats include full precision binary (for spreadsheets such as Excel) and ASCII (for pasting into text editors). All information will be present, regardless of the number of eigenmodes reconstructed.

Save



The **Save Data to Disk** option writes the x-values and the output currently plotted to a supported file format. These formats include ASCII, Excel 97/2000, Excel 95, Lotus WK3, Lotus WK1, SPSS, or Systat. All information will be present, regardless of the number of eigenmodes reconstructed.

Production Facility



The **TableCurve 2D Automation** facility allows unattended processing of large numbers of data sets. The data sets can be consolidated in an Excel file or acquired using a DLL. The graphs and reports can be exported to an MS Word/RTF file, while the processed data can be exported to an Excel 95 or Excel 97/2000 file.



The **Reset** button restores the data to its state when first entering the procedure. If an **Automation Session** is in progress, the **Reset** button can be used to terminate the automated processing.

Residuals



The **Residuals** button is used to graphically display the difference between the original and filtered data.

Numeric Summary



The **Numeric Summary** offers an eigendecomposition report. The report includes a table of eigenvalues, normalized eigenvalues, and partial r^2 correlation coefficients.

MS Word/RTF Export



The **MS Word/RTF File Export** option is used to save the current graphs and reports to either an MS Word file or a portable RTF (Rich Text Format) file. The graphs are inserted into the file as Windows metafiles.

Automatic Update

The List, Residuals, and Numeric Summary windows are automatically updated when any change is made to the algorithm's settings.

Updating the Data Table

 When exiting this procedure with the **OK** button, an option will be presented to update TableCurve 2D's main data table with the filtered data.

If **Background Thread Fitting** is active, the fitting will be initiated as soon as the filtering is accepted and the data are rescanned.

Fourier Processing

Fourier Decomposition

The Fast Fourier Transform (FFT) decomposes a time-domain signal (which can be a function of time, spatial coordinates, or any time series abscissa) into complex exponentials (sines and cosines). A Fourier transform offers a complete picture of frequency space, but retains no information as to when in time a signal occurs. Thus a signal should either be wide-sense stationary, with a constant mean and variance across broad time segments, or you must care only qualitatively about whether a certain frequency content signal occurs somewhere in the time range being sampled.

The FFT is a fast algorithmic route for producing the Discrete Fourier Transform (DFT). The forward and reverse discrete transforms are defined as follows:

$$X_n = \sum_{k=0}^{N-1} x_k e^{\frac{i2\pi kn}{N}}$$

$$x_k = \frac{1}{N} \sum_{n=0}^{N-1} X_n e^{\frac{-i2\pi kn}{N}}$$

While the DFT is very simple, it is an order n^2 procedure whereas the FFT is an $n \log_2(n)$ operation. The difference in processing times is dramatic with large data sets. TableCurve 2D offers four different **FFT algorithms**.

Fourier decompositions are limited in resolution, as the frequencies at which the sines and cosines are computed are equally spaced and fixed in number. For a data set of n points, there will be n complex frequencies. For real data, the negative frequencies mirror the positive ones, and only the positive frequencies are displayed. Thus, $n/2+1$ frequencies comprise the spectrum. Normalized frequencies range from 0, sometimes referred to as DC, to 0.5, the Nyquist frequency, the maximum frequency that can still be detected given the selected sampling rate. TableCurve 2D displays the actual frequencies. Normalized frequencies are not used.

Continuous Basis Functions

The Fourier decomposition represents a data sequence as a linear combination of a set of sine and cosine basis functions. Although the data and Fourier sequences are each discrete, the basis functions are continuous and infinite in duration. It is thus possible to reconstruct the signal for any time within the range of the original sequence:

$$y(t) = \sum_{k=1}^{N_{spec}} A_k \sin(2\pi \nu_k t + \theta_k)$$

The Fourier basis functions can be treated as complex exponentials, zero phase sine and cosine pairs, phase bearing cosines, or phase bearing sines. In TableCurve 2D, the Fourier basis functions are reported as phase bearing sinusoids. In the above equation, **A** is the amplitude reported, **nu** is the

frequency, and **theta** is the phase. The signal at any time **t** can be reconstructed by summing the value of all **Nspec** sinusoids in the spectrum evaluated at that time. The amplitudes are adjusted so that all power is represented in the positive frequencies.

This reconstruction is the basis of the **Fourier Estimation** option in the Estimate menu.

Spectral Leakage

Spectral leakage is the term used to describe the loss of power of a given frequency to other frequency bins in the FFT. Approximating a finite data stream with an infinite Fourier series assumes the data are fully periodic. Few data sets actually evidence this continuous wraparound property. Instead there are edge effects arising from the discontinuity at the bounds that cause spectral leakage.

Spectral leakage also occurs as a consequence of the linear fixed frequencies used in the FFT. Unless a sinusoid's frequency just happens to correspond with the exact center of an FFT frequency bin, some portion of the signal will leak to other bins. One can expect, therefore, some portion of the adjacent bins to also reflect some measure of this peak. In fact, when a spectral component's frequency is not centered in a bin, some portion of the variance due to this component may be accounted for by each bin in the spectrum.

This tends to diminish as you get further away from the frequency of interest, although there will be harmonics. A sine and cosine at multiples of the original frequency, for example, would be expected to capture some of the variance as well. From another viewpoint, an FFT can be considered a linear least-squares fit to sine and cosine functions at fixed linearly increasing frequencies. If a single sine wave is fit non-linearly, the parameters can be fully resolved since the amplitude, frequency, and phase can each vary in order to find the optimum. This is not possible in an FFT. While the amplitude and phase can vary, the frequencies cannot. Unless the frequency of the sine wave just happens to fall exactly at one of the FFT's fixed frequencies, the sine component will be smeared across multiple frequencies.

While the spectral leakage attributable to the fixed frequencies is an intrinsic part of a Fourier spectrum, much can be done to eliminate edge effects. To produce the minimum spectral leakage, the data can approximate an infinite sequence by decaying to zero at both endpoints. This is accomplished by using a data tapering window.

Data Tapering Windows

These windows are generally applied in the time domain, even when they are built within the frequency domain. This is because a simple multiplication is all that is needed to implement the tapering in the time domain. TableCurve 2D includes a broad selection of **data tapering windows** with merit in signal processing.

A rectangular window (no actual window is applied) produces a one-sided bin width of 1.0 in the frequency domain. The data tapering windows in TableCurve 2D vary in Fourier bin width from 1.1 to 6.0. Many popular windows have widths of 2.0, 3.0, and 4.0. When a data set is windowed, portions of the signal nearer the bounds are discarded, and resolution is degraded. A window with a bin width of 2.0 will have only one-half the resolution of its non-windowed counterpart. A window with a bin width of 4.0 will have only one-quarter the resolution.

Although some portion of the data's influence is discarded or diminished, and despite the loss of resolution, windowed FFTs are often the best course for accurate estimation of spectral components, especially when a high dynamic range is needed. By having the data smoothly vary and decay to zero at the endpoints, spectral leakage is drastically reduced, so much so that a decibel (dB) scale, which is logarithmic, must be used to observe the spectral leakage across the bins in a meaningful way.

When an FFT processes data that have been tapered by a time domain window, each frequency bin effectively becomes wider. This width is sometimes referred to as the order of the window. **Hann** and

Hamming windows spread the signal across 2 bins and are thus order 2 windows. This spreading eliminates much of the spectral leakage associated with being too near an edge of a frequency bin.

While there are a variety of window taper properties, the bin width is certainly the most important since this determines the resolution of the signal as well as the dynamic range the FFT is capable of representing. Each window has a characteristic shape which fixes most of the resolving properties of the resultant frequency spectrum. Some windows have an adjustable parameter, and for all of TableCurve 2D's data tapering windows, this controls the width of the peak in the frequency domain. The control for most adjustable windows has been modified to directly reflect this bin width.

The **Chebyshev** window is an exception. It produces the narrowest possible peak for a specific side lobe (adjacent harmonic) leakage level. Constructed in the frequency domain, the adjustable control specifies this maximum side lobe level directly.

Certain windows minimize the leakage at adjacent sidelobes, at the cost of low to negligible frequency decay (rolloff) as you move further away from the peak in frequency. These windows give excellent resolution of closely spaced peaks of varying power in the spectrum, but may mask peaks of low power far from the peak frequencies.

Other windows maximize the decay or rolloff of the spectral leakage, but are not as good at minimizing the leakage at nearby frequencies. Two peaks distant in frequency, each of low power, may be well detected, but in the case of two closely spaced peaks, one of greater power may fully mask one of lesser power.

There are thus two key design criteria in selecting a data tapering window. The first is resolution. Select the greatest bin width that offers acceptable resolution. The second involves selecting a window of that order based on whether or not you want to primarily resolve distant peaks of low power (where a maximum rolloff window is selected), or whether a low power peak may be in the shadow of a nearby component of greater power. Often a compromise is chosen. The best-known windows, such as the popular 4-parameter Blackman-Harris, **cs4 BHarris min**, tend to be those which minimize the adjacent sidelobe spectral leakage.



It is not easy to commit the properties of a given window to memory, and they are important in using a windowed FFT effectively. To simplify data tapering window selection, TableCurve 2D offers the means to evaluate the frequency domain performance in the **Explore Data Tapering Windows** option.

Data Tapering Windows

Data tapering windows are used to reduce spectral leakage in Fourier spectra.



Data tapering windows can be evaluated for frequency domain performance in the [Explore Data Tapering Windows](#) option.

The following fixed shape data tapering windows are available in TableCurve 2D:

- None
- Welch
- Bisquare
- Bartlett
- cs2 Hann
- Tukey-Hann
- cs2 Hamming
- Bartlett Mod
- cs3 Nuttall C3
- cs3 Blackman
- cs3 BHarris 3
- cs3 Nuttall C1
- cs3 BlckmnExct
- cs3 BHarris min
- cs3 Nuttall min
- cs4 Nuttall C5
- cs4 BHarris 4
- cs4 Nuttall C3
- cs4 Nuttall C1
- cs4 BHarris min
- cs4 Nuttall min

The following adjustable shape data tapering windows are available in TableCurve 2D:

- Beta
- csx maxRolloff
- Kaiser-Bessel
- VanderMaas
- Chebyshev
- Chebyshev Appr
- Slepian DPSS
- Gaussian
- Tapered-Cosine

Welch

$$1.0 - (((n-1)-2*i)/(n-1))*((n-1)-2*i)/(n-1), i=0..n-1$$

Bisquare

$$(1.0 - (\text{abs}(i - 0.5*n + 0.5))^2 / (0.5*n - 0.5)^2)^2, i=0..n-1$$

Bartlett

$2*i/(n-1)$, $i=0..(n-1)/2$
 $2-2*i/(n-1)$, $i=(n-1)/2+1..n-1$

cs2 Hann

$0.5-0.5*\cos(2*\pi*i/(n-1))$, $i=0..n-1$

Tukey-Hann

$\text{Dirichlet}(\theta)=\sin((2*\pi+0.5)*\theta)/(2*\pi*\sin(0.5*\theta))$
 $0.25*\text{Dirichlet}(2*(i/(n-1))-1.0-0.5)+(1.0-2.0*0.25)*\text{Dirichlet}(2*(i/(n-1))-1.0)+0.25*\text{Dirichlet}(2*(i/(n-1))-1.0+0.5)$, $i=0..n-1$

cs2 Hamming

$0.53836-0.46164*\cos(2*\pi*i/(n-1))$, $i=0..n-1$

Bartlett Mod

$(\sin(\pi*(2*(i/(n-1))-1)))^2/(2*\pi*\sin((i/(n-1))-0.5))^2$, $i=0..n-1$

cs3 Nuttall C3

$0.375-0.5*\cos(2*\pi*i/(n-1))+0.125*\cos(4*\pi*i/(n-1))$, $i=0..n-1$

cs3 Blackman

$0.42-0.5*\cos(2*\pi*i/(n-1))+0.08*\cos(4*\pi*i/(n-1))$, $i=0..n-1$

cs3 BHarris 3

$0.44959-0.49364*\cos(2*\pi*i/(n-1))+0.05677*\cos(4*\pi*i/(n-1))$, $i=0..n-1$

cs3 Nuttall C1

$0.40897-0.5*\cos(2*\pi*i/(n-1))+0.09103*\cos(4*\pi*i/(n-1))$, $i=0..n-1$

cs3 BlackmanExct

$0.42659071367153912296-0.49656061908856405847*\cos(2*\pi*i/(n-1))+0.076848667239896818573*\cos(4*\pi*i/(n-1))$, $i=0..n-1$

cs3 BHarris min

$0.42323-0.49755*\cos(2*\pi*i/(n-1))+0.07922*\cos(4*\pi*i/(n-1))$, $i=0..n-1$

cs3 Nuttall min

$0.4243801-0.4973406*\cos(2*\pi*i/(n-1))+0.0782793*\cos(4*\pi*i/(n-1))$, $i=0..n-1$

cs4 Nuttall C5

$0.3125-0.46875*\cos(2*\pi*i/(n-1))+0.1875*\cos(4*\pi*i/(n-1))-0.03125*\cos(6*\pi*i/(n-1))$, $i=0..n-1$

cs4 BHarris 4

$0.40217-0.49703*\cos(2*\pi*i/(n-1))+0.09892*\cos(4*\pi*i/(n-1))-0.00188*\cos(6*\pi*i/(n-1))$, $i=0..n-1$

cs4 Nuttall C3

$0.338946 - 0.481973 \cos(2\pi i/(n-1)) + 0.161054 \cos(4\pi i/(n-1)) - 0.018027 \cos(6\pi i/(n-1)), i=0..n-1$

cs4 Nuttall C1

$0.355768 - 0.487396 \cos(2\pi i/(n-1)) + 0.144232 \cos(4\pi i/(n-1)) - 0.012604 \cos(6\pi i/(n-1)), i=0..n-1$

cs4 BHarris min

$0.35875 - 0.48829 \cos(2\pi i/(n-1)) + 0.14128 \cos(4\pi i/(n-1)) - 0.01168 \cos(6\pi i/(n-1)), i=0..n-1$

cs4 Nuttall min

$0.3635819 - 0.4891775 \cos(2\pi i/(n-1)) + 0.1365995 \cos(4\pi i/(n-1)) - 0.0106411 \cos(6\pi i/(n-1)), i=0..n-1$

Beta

alpha=main lobe width

$(4*(1-i/(n-1))^i/(n-1))^{(-3.218913776512187+2.760793796409310*\alpha)}, i=0..n-1$

csx maxRolloff

alpha=main lobe width

This window produce the maximum rolloff characteristics.

$(\text{abs}(0.5*(1-\cos(2\pi i/(n-1))))^{(\alpha-1)}, i=0..n-1$

Kaiser-Bessel

alpha=main lobe width, I0 is Modified Bessel

$I_0(\sqrt{((\alpha*\alpha-1.0)/0.10132118361)*\sqrt{i*(2*((n-1)/2)-i))/((n-1)/2)})/I_0(\sqrt{((\alpha*\alpha-1.0)/0.10132118361)}), i=0..n-1$

VanderMaas

alpha=main lobe width, I1 is Modified Bessel

$0.5*I_1(2*(0.5*\pi*\sqrt{4*\alpha*\alpha-1})*\sqrt{(i/(n-1))*(1-i/(n-1))})/(\sqrt{(i/(n-1))*(1-i/(n-1))})*I_1(0.5*\pi*\sqrt{4*\alpha*\alpha-1})), i=0..n-1$

Chebyshev

This window produces the minimum sidelobe leakage, but offers zero rolloff. It is constructed in the frequency domain and inverted to produce the time domain data. For more information, see program 5.2 in IEEE Programs for Digital Signal Processing, IEEE Press, 1979. Unlike the alpha in the other adjustable windows, the Chebyshev adjustable parameter sets the sidelobe level in -dB. TableCurve 2D offers the Chebyshev window from -30dB to -150dB.

Gaussian

alpha=main lobe width

$\exp(-4*\ln(2)*((i-((n-1)/2))/((n-1)*((2.453274901281656+\alpha*0.3202556336455866)/(1.0+\alpha*(2.348619671799226))))))^2, i=0..n-1$

Tapered-Cosine

taper ranges from Hann with alpha=0.5 to rectangular as alpha approaches 0

$\text{if}(i > (\text{int})(\alpha*n+0.5) \text{ and } i < (\text{int})((1.0-\alpha)*n-0.5)), 1.0, 0.5-0.5*\cos((\pi/\alpha/(n-1))*(i-0.5*n, n-i-1, i))), i=0..n-1$

The following reference may be of interest: Albert H. Nuttall, "Some Windows with Very Good Sidelobe Behavior", IEEE Trans. ASSP, v29-1, Feb. 1981.



Explore Data Tapering Windows

Data tapering windows are used to reduce the spectral leakage in Fourier spectra. Since it can be difficult to commit the properties of the various windows to memory, especially when they contain an adjustable parameter, and since they are important in using a windowed FFT effectively, TableCurve 2D offers this [Explore Data Tapering Windows](#) option.

This option plots a discrete FFT in a problem designed to illustrate the frequency widths of each window as well as the rolloff vs. maximum sidelobe tradeoff. You can view as many as three windows simultaneously. Both the time and frequency domain plots are displayed. A numerical display of the three key properties of tapering windows is also given for the first window.

FFT n and Win n

In this posed problem, the data tapering window of length [Win n](#) is constructed at the center of a data series of length [FFT n](#). The default values use a data length of 1001 with a window length of 101 in the center of the data stream. The remaining 900 values are 0. An exact n FFT is made to produce the corresponding frequency spectra.

Win Cnt

Inspect the spectra for up to 4 windows simultaneously.

Window Selection

For each data tapering window up to the count specified, select a window, and if it includes an adjustable parameter, you must also set it.

Window Properties

- [Mainlobe](#) - the one-sided normalized frequency width from 0 frequency to the first null.
- [Sidelobe](#) - the delta in dB between signal at 0 frequency and the highest sidelobe signal. The highest sidelobe is usually, but not always, adjacent.
- [Rolloff](#) - This value in dB/octave represents the whole of the decay in the frequency domain, rather than the asymptotic rolloff. A non-linear robust (Lorentzian minimization) log fit is used to determine this value.

Note that these are empirical values derived directly from the frequency spectrum. You should find good agreement with analytic values reported in the literature.

FFT Algorithms

TableCurve 2D offers four different FFT algorithms and a composite procedure that minimizes computation time.

FFT Radix 2

This is the conventional power of 2 FFT procedure. TableCurve 2D supports the following n : 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, and 65536. If the data size is not a power of 2, zeros are appended to the next larger n . To produce the fastest possible power of 2 FFTs, TableCurve 2D uses the Intel Signal Processing Library. This set of processor-specific DLLs includes architecture optimized FFTs. This is the fastest of the FFT algorithms.

Prime Factor

The prime factor FFT procedure used in TableCurve 2D processes primes through 13. TableCurve 2D uses an implementation of Temperton's Prime Factor FFT (C. Temperton, "Implementation of a Self-Sorting In-Place Prime Factor FFT Algorithm", Journal of Computational Physics, v. 58, p. 283, 1985). For real data, the following n are processed exactly: 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 36, 40, 42, 44, 48, 52, 56, 60, 66, 70, 72, 78, 80, 84, 88, 90, 96, 104, 110, 112, 120, 126, 130, 132, 140, 144, 154, 156, 160, 168, 176, 180, 182, 198, 208, 210, 220, 224, 234, 240, 252, 260, 264, 280, 286, 288, 308, 312, 330, 336, 352, 360, 364, 390, 396, 416, 420, 440, 462, 468, 480, 504, 520, 528, 546, 560, 572, 616, 624, 630, 660, 672, 720, 728, 770, 780, 792, 840, 858, 880, 910, 924, 936, 990, 1008, 1040, 1056, 1092, 1120, 1144, 1170, 1232, 1248, 1260, 1320, 1386, 1430, 1440, 1456, 1540, 1560, 1584, 1638, 1680, 1716, 1760, 1820, 1848, 1872, 1980, 2002, 2016, 2080, 2184, 2288, 2310, 2340, 2464, 2520, 2574, 2640, 2730, 2772, 2860, 2912, 3080, 3120, 3168, 3276, 3360, 3432, 3640, 3696, 3744, 3960, 4004, 4290, 4368, 4576, 4620, 4680, 5040, 5148, 5280, 5460, 5544, 5720, 6006, 6160, 6240, 6552, 6864, 6930, 7280, 7392, 7920, 8008, 8190, 8580, 8736, 9240, 9360, 10010, 10080, 10296, 10920, 11088, 11440, 12012, 12320, 12870, 13104, 13728, 13860, 14560, 15840, 16016, 16380, 17160, 18018, 18480, 18720, 20020, 20592, 21840, 22176, 22880, 24024, 25740, 26208, 27720, 30030, 32032, 32760, 34320, 36036, 36960, 40040, 41184, 43680, 48048, 51480, 55440, 60060, 65520. For other sizes, zeros are appended to the next larger n . This is the second fastest of the FFT procedures.

Mixed Radix

The mixed radix FFT algorithm processes any size n exactly. TableCurve 2D uses a modification of Singleton's mixed radix algorithm (R. C. Singleton, "An Algorithm for Computing the Mixed Radix Fast Fourier Transform", IEEE Trans. Audio Electroacoust., v. AU-17, p. 93, June 1969) that removes the size limitation. This algorithm is slower than the radix 2 and prime factor procedures.

Chirp-Z

The Chirp-Z FFT algorithm (L. R. Rabiner, R. W. Schafer, C. M. Rader, "The Chirp z-Transform Algorithm and Its Application", BSTJ, 48, p.1249, May-June 1969) can also be used to exactly process any size n . The TableCurve 2D Chirp-Z implementation computes the z-transform at n equally spaced points on the unit circle, reproducing the FFT. The fast convolution requires two radix-2 FFT forward transforms and 1 FFT inverse. When very large primes are present, this algorithm is faster than the mixed radix procedure.

Best Exact N

This is the recommended algorithm for most procedures within TableCurve 2D that use the FFT. If the data size is a power of 2, the [FFT Radix 2](#) algorithm is used. If not and the size is included in the prime-factor set, then the [Prime Factor](#) procedure is used. Otherwise, the [Mixed Radix](#) algorithm is used if the largest prime ≤ 509 and the [Chirp-Z](#) is used if the largest prime ≥ 521 . This produces the fastest possible exact- n FFT.

Numeric Summary (FFT)

This option offers a full FFT report.

Initial Information

The following information is always included:

- Main, x, and y data titles
- File source for data
- Report date
- FFT algorithm selected
- Data tapering window used (if present)
- Size of data
- Size of FFT and prime factorization
- Total power of spectrum, SSA, MSA, TISA
- Signal count detected
- Linear trend removal state

Interpolated Spectral Peaks

The first optional item, available from the Format menu, is [Add Interpolated Spectral Peaks](#). The bin-interpolated peaks, based upon the signal count, are listed. This table will consist of interpolated spectral frequencies and interpolated values for whichever spectral quantity is being plotted. Note that the [dB Norm](#) format normalizes to 0 dB with the actual spectral data series. The peak corresponding to this 0 dB channel will typically report a positive dB value because of the bin interpolation.

Frequency Analysis

The second optional item is [Add Frequency Analysis](#). This table also uses interpolated frequencies and computes interpolated amplitudes and phases (sine-based). For each peak, the TISA power (based on amplitude) is computed. Note that spectral leakage to other bins will reduce the power reported for each component.

Absolute and relative percents are also given. These are often the quantities of interest when comparing strengths of signal components. The summed power reported in the frequency analysis is merely the sum of the component powers. It is not the power of the composite signal that would result from the addition of the components. In most instances, this sum will be lower than the TISA power of the incoming data.

Linear Sine Component Fit

When the [Add Sine Component Linear Fit Summary](#) item in the Format menu is checked, a linear sinusoidal least-squares fit is made to the incoming data. The report will then include a summary of this suboptimal sinusoidal parametric regression. In this linear fit, the sinusoid count is set to the number of spectral peaks identified in the procedure and the frequencies are locked at the values determined from the spectrum. For this procedure, the frequencies will derive from the interpolated values found from the FFT. In general, an FFT is not a particularly accurate frequency estimator even with a good bin interpolation algorithm. Further, the limited spectral resolution of the FFT may result in more than one component appearing as a single spectral peak.

Since the amplitudes and phases are determined from the sinusoidal parametric model, the influence of spectral leakage is no longer a factor.

If the [Add Sine Component Linear Fit Details](#) item is checked, the report will also include a statistical breakdown of each sinusoid in the fit.

Eigendecomposition

Eigendecompositions are used in the following algorithms within TableCurve 2D. They are the principal methodology in the:

- [Eigendecomposition Denoising](#) (Filter menu)
- [Eigendecomposition Reconstruction](#) (Filter menu).

Eigendecompositions are also intrinsically a part of all TableCurve 2D procedures that include SVD (singular value decomposition) methods:

- [AR Modeling and Prediction](#) (Estimate menu)

Processing the eigenmodes (eigenvectors and eigenvalues/singular values) of a data series is an important tool in signal analysis. Unlike Fourier decompositions, which partition signals based on harmonic frequency using parametric sines and cosines, an eigendecomposition partitions by signal strength using adaptive non-parametric basis functions. Signal components can thus be separated by differences in power.

Nomenclature

The identification, isolation, and reconstruction of signal components via eigendecomposition is known by a variety of names. "Singular Spectrum Analysis", "Principal Component Analysis", and "Eigenfiltering" are common. TableCurve 2D exclusively uses the "Eigendecomposition" designation since it represents a more precise description of the numeric method.

Reference

An excellent reference for eigendecomposition can be found in:

- J. B. Elsner and A. A. Tsonis, "Singular Spectrum Analysis", Plenum Press, 1996.

Eigenvalues and Singular Values

Although the terms eigenvectors and singular vectors are often used interchangeably, the singular values are the square roots of the eigenvalues. In terms of visually selecting eigenmodes for signal-noise separation, or for component isolation, either the eigenvalues or their square root, the singular values, can be used. For [signal-noise thresholding](#), TableCurve 2D plots the singular values directly. The [Eigendecomposition Denoising](#) and [Eigendecomposition Filtering](#) procedures plot the eigenvalue (the square of the respective singular value) divided by the order of the decomposition.

Lagged Covariance or Data Matrix

An eigendecomposition can be achieved in any number of ways. The first step is always the creation of a matrix that uses lagged copies of subsets of the data series. This can be a straightforward data or trajectory matrix, such as the forward prediction ([Fwd](#)), backward prediction ([Bwd](#)), or forward-backward ([FB](#)) prediction matrices used in [autoregressive modeling](#). These data matrices are usually rectangular, and SVD is used to extract the eigenvectors and singular values. If the eigendecomposition is used only for reconstruction and does not involve the least-squares computation of parametric model coefficients, there is no difference between using a [Fwd](#) or [Bwd](#) data matrix. A [FB](#) prediction matrix will contain twice the number of rows as a [Fwd](#) or [Bwd](#) matrix, and will require a greater processing time. This approach is used in the [AR Modeling and Prediction](#) procedure.

Another option is to use one of several methods to construct a covariance matrix from lagged copies of the data. This is a square matrix whose eigenvectors and singular values can be computed using SVD. A covariance-matrix procedure that enforces Toeplitz symmetry (all of the elements along each diagonal are equal), is reported to be of value for short data records. This analysis imposes a stationary assumption. Typically, non-Toeplitz matrices will better map the variance in a data series. The covariance

matrix approach is used in the [Eigendecomposition Denoising](#) and [Eigendecomposition Filtering](#) options.

Eigendecomposition Order

The order of the eigendecomposition is the number of data elements from the data set that is extracted for each segment or subset of the overall series, not the number of segments. For rectangular data matrices, the order or "embedding dimension" specifies the number of columns in the matrix and the number of segments specifies the rows. For a covariance matrix, both the column and row count will be equal to the order.

It is essential that the order of the eigendecomposition be high enough to offer good signal-noise separation. The order must be sufficient to fully partition the noise components and prevent them from corrupting the signal-being eigenmodes. In general, the higher the order (data length permitting), the more complete the partitioning as more eigenmodes are available to capture the random noise.

Denoising

An inverse FFT can reconstruct a filtered version of the original data by zeroing a subset of the frequency channels. Similarly an eigendecomposition can reconstruct a filtered data set by discarding one or more eigenmodes. The most obvious eigendecomposition-based filtering is for the removal of random noise from a deterministic signal. Noise removal via eigendecomposition is the sole purpose of the [Eigendecomposition Denoising](#) procedure. Unlike Fourier low-pass filtering, where some noise passes in the low frequency channels, and time-domain smoothing, which can introduce spurious spectral features, eigendecomposition procedures can theoretically isolate and discard virtually all of the noise corrupting a given signal, across all frequencies.

Isolating Signal or Noise Components

The Eigendecomposition Filtering and Reconstruction option can be used to isolate individual signal components based on signal strength. The first eigenmode captures the greatest measure of the variance in the data; the least eigenmode usually captures only the least significant noise oscillations. In general, eigendecompositions cannot isolate a single sinusoid amongst a number of sinusoids when signal strengths are approximately comparable.

The first eigenmode will capture the predominant data trend in the signal, the second the next predominant, and so forth. It does not matter if the component captured is sinusoidal, a square wave, a sawtooth, or an anharmonic pattern. Further, the signal may be a slowly varying low frequency anharmonic oscillation, or a high frequency sinusoid. The eigenmodes are said to be adaptive, because they capture, in an eigenvalue-ordered sequence, the variance of the data in a non-parametric manner.

Two eigenmodes are need to capture an oscillatory trend. A pair of eigenmodes with nearly identical eigenvalues usually signifies a specific harmonic or anharmonic oscillation in the signal. One strength of eigendecomposition is the isolation of secondary signal components containing very little power. Such secondary components may not be visible in Fourier spectra because of spectral leakage and resolution considerations. Similarly these components may be invisible in AR and ARMA models because an autoregressive model will emphasize the primary components. The same is true for parametric sinusoid models because the total variance associated with the components of greater power can often overwhelm a least-squares fit (and even a robust fit) that seeks to also include secondary components of far lesser power. Eigendecomposition offers the means to isolate the high and low power components for separate spectral analysis and fitting.

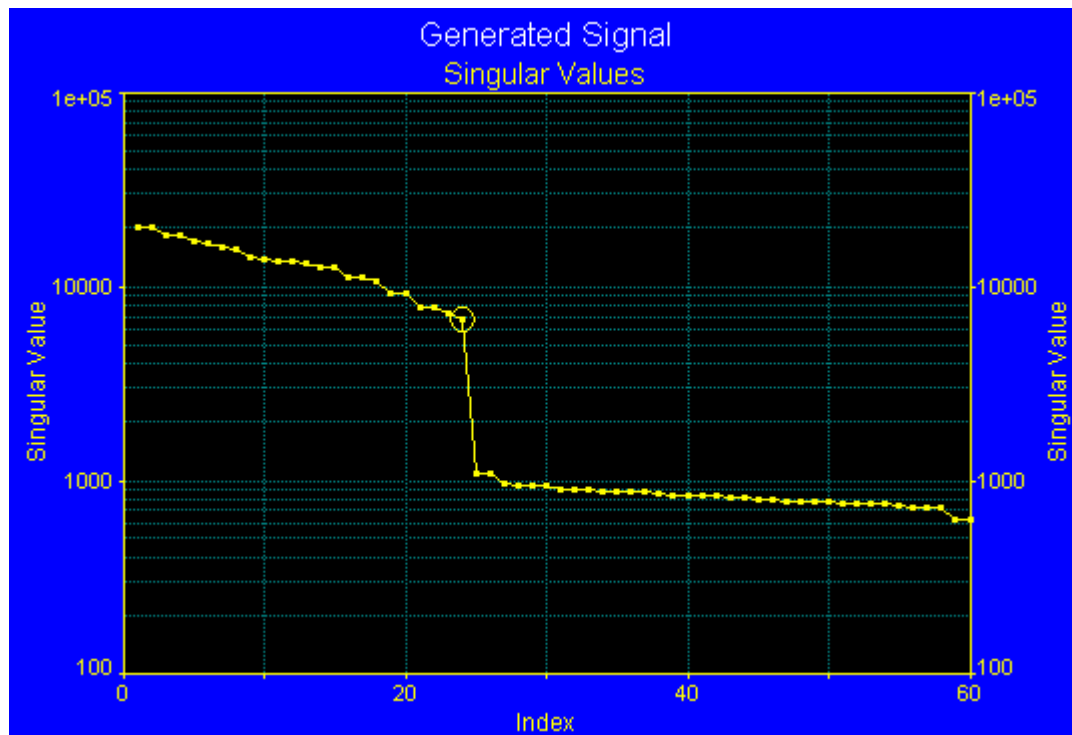
With eigendecomposition, it is possible to isolate backgrounds by reconstructing only the noise eigenmodes. This can be done to confirm the presence or absence of white (normally distributed) noise, or to isolate a red noise background for subsequent modeling by one of TableCurve 2D's spectral procedures. It is also possible to reconstruct only the initial noise components, those judged to contain no meaningful signal, and to check for oscillatory trends in the time domain, or for significant spectral content in the frequency domain.

Eigendecomposition Signal/Noise Threshold

Procedures that perform an **eigendecomposition**, either for data reconstruction or solving for parametric coefficients, offer the means to graphically select the number of eigenmodes that are to be included in the signal space. The retained eigenmodes should represent the signal or principal components since these are used for the procedure's solution or reconstruction. The discarded eigenmodes should represent the noise since this information is zeroed and does not factor into the solution. This signal-noise separation is intrinsic to SVD procedures.

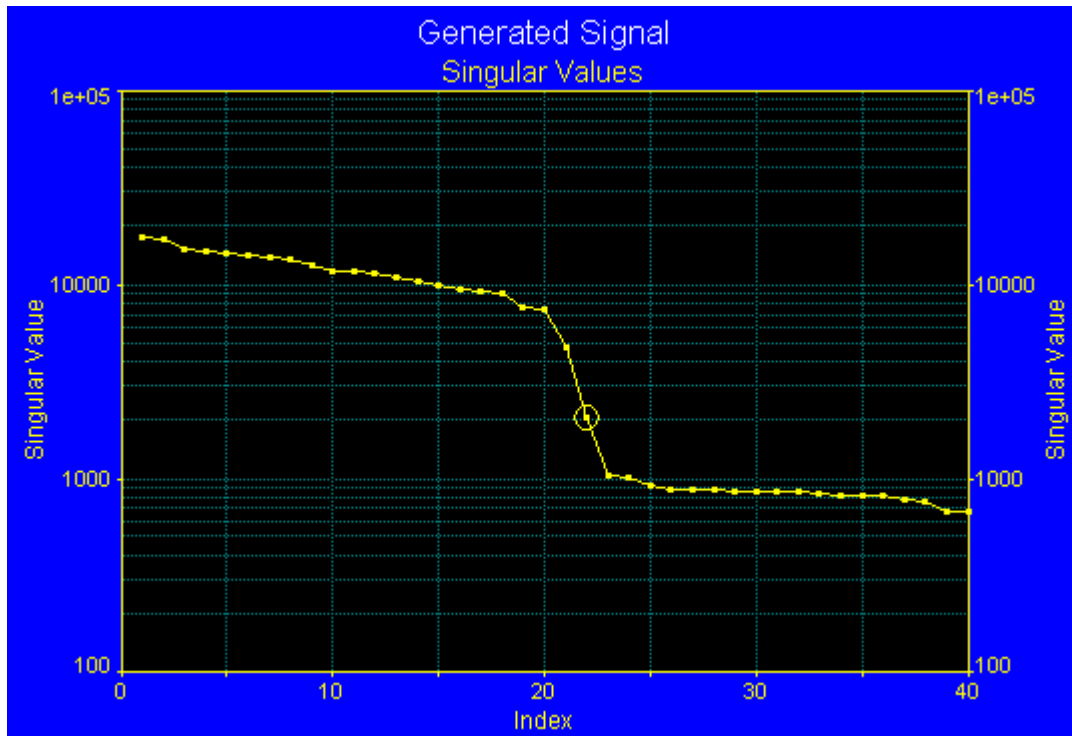
When a sufficient matrix order exists to effect this signal-noise separation, and provided signal strengths and properties are such that an **eigendecomposition** can isolate signal components, the singular value or eigenvalue plot will reveal a threshold that reveals a clear demarcation between signal and noise. The eigenvalues or singular values usually require a log scale in order to perceive this transition. Since the singular values are the square root of the eigenvalues, the eigenvalues will show a greater separation in relative magnitude.

To select the optimum signal space, left click the mouse on the last eigenvalue or singular value that represents signal. This eigenmode and all prior ones will be processed while all subsequent singular values will be zeroed. The example below contain 12 sinusoids and white noise. With an AR model order of 60, the Data SVD FB procedure produces the following signal-noise threshold. Since two eigenmodes are needed to catch each oscillatory signal component, 24 singular values are needed to fully capture the signal. The selected point is the 24th eigenmode and the transition is clean.

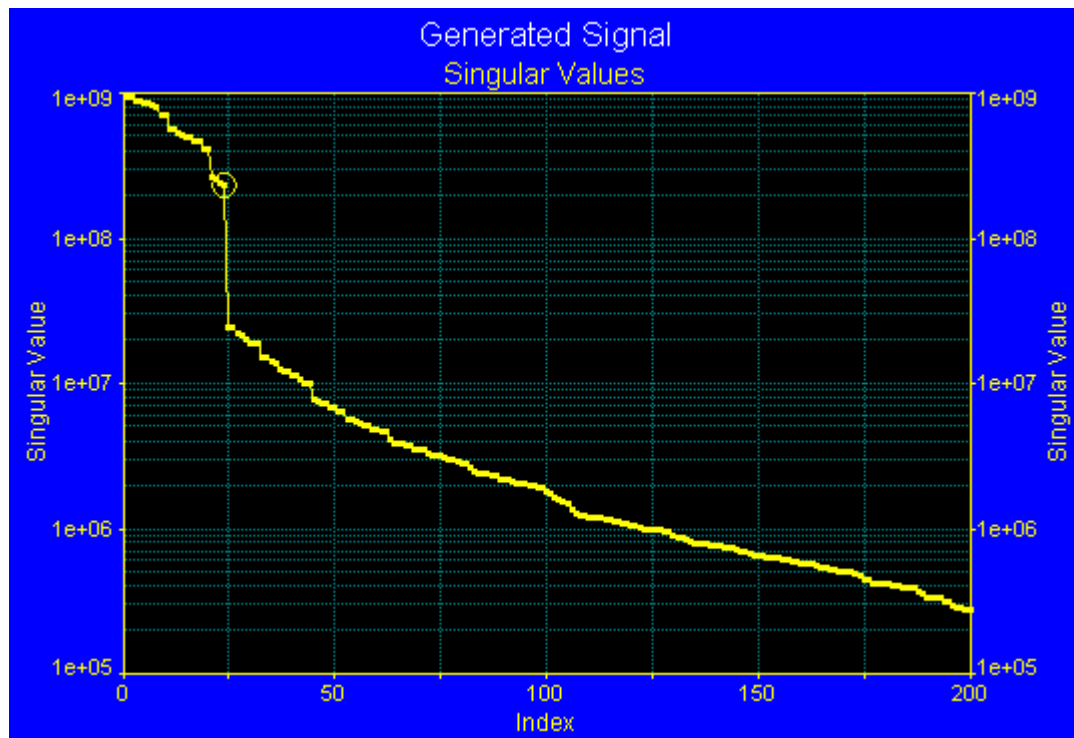


The only harm in fitting very high orders with SVD is the increased processing time. In general, you need to strike a balance between SVD processing time and model order. With SVD, you must choose a high enough model order to threshold the signal and noise effectively. A transition, of itself, is not an indicator of a sufficient model order. The singular value plot below is for this same example, except that an AR

model order of 40 was used. Note the two eigenmodes that occur in the transition band. Here the 22nd eigenmode represents the last signal element before the noise floor. The model order is insufficient.



If you can tolerate the processing time and have sufficient free memory available, a very high order can be safely used. The higher the order, the more complete the signal-noise thresholding can be. The Eigendecomposition options can use a square covariance matrix which readily supports processing large data sets with high model orders. Similarly, the AR procedure offers SVD normal equations algorithms which are appreciably faster with large data sets than processing the full data matrix. The following singular value plot is for a 200th model order **Nrml SVD FB AR** fit.



White (Gaussian distributed) noise is most easily thresholded using SVD. Red noise, which decreases in power as frequency increases, is not as readily partitioned using the SVD. Also, bear in mind that all eigendecomposition procedures partition based upon signal strength. When noise is present at levels that start to approach that of the signal, the characteristic transition may be impossible to achieve. Fitting the highest order for a given algorithm is one way to confirm whether or not this signal-noise transition can be achieved.



Signal Threshold Selection

All options which use **SVD** (Singular Value Decomposition) for solving matrices offer this option for graphically selecting the number of eigenmodes that are to be included in the signal space. The retained eigenmodes should represent the signal or principal components since these are used for the procedure's solution. The discarded eigenmodes should represent the noise since this information is zeroed and does not factor into the solution. This signal-noise separation is intrinsic to SVD procedures.

Singular Value Plot

When a sufficient matrix order exists to effect this **signal-noise separation**, and provided signal strengths and properties are such that **eigendecomposition** can isolate signal components, the singular value plot will reveal a threshold that reveals a clear demarcation between signal and noise. The singular values usually require a log scale in order to perceive this transition.

Thresholding

To select the **optimum signal space**, simply left click the mouse on the last singular value that represents signal. This eigenmode and all prior ones will be processed while all subsequent singular values will be zeroed. The only harm in fitting very high orders with SVD is the increased processing time. In general, you will need to strike a balance between SVD processing time and model order. With SVD, you must choose a high enough model order to **threshold the signal and noise** effectively.

Model Order Considerations

If you can tolerate the processing time and have sufficient free memory available, a very high order can be safely used. The higher the order, the more complete the signal-noise thresholding can be. The **AR Modeling and Prediction** procedure offers **SVD normal equations algorithms** which are appreciably faster with large data sets than processing the full data matrix.

Noise Types

White (Gaussian distributed) noise is most easily thresholded using SVD. Red noise, which decreases in power as frequency increases, is not as readily partitioned using the SVD. Also, bear in mind that all **eigendecomposition** procedures partition based upon signal strength. When noise is present at levels that start to approach that of the signal, the characteristic transition may be impossible to achieve. Fitting the highest order for a given algorithm is one way to confirm whether or not this signal-noise transition can be achieved.

Lack of Visual Threshold

When this transition cannot be achieved, you can use the **Eigendecomposition Filtering** option to isolate components by selecting pairs of nearly equal magnitude singular values. It may also be possible to determine from reconstructing select eigenmodes where noise finally overwhelms signal elements.

List



The **List Data** option lists a table of the singular values. The listing uses the **TableCurve 2D text viewer** facility.

Copy



The **Copy Data to Clipboard** option copies the singular values to the clipboard. Formats include full precision binary (for spreadsheets such as Excel) and ASCII (for pasting into text editors).

Save



The [Save Data to Disk](#) option writes the singular values to a supported file format. These formats include ASCII, Excel 97/2000, Excel 95, Lotus WK3, Lotus WK1, SPSS, or Systat.

Numeric Summary (Eigendecomposition)

This option offers an eigendecomposition analysis for the [Eigendecomposition Denoising](#) and [Eigendecomposition Filtering](#) options.

Initial Information

The following information is always included:

- Report date
- Main, x, and y data titles
- File source for data
- Data size
- Algorithm use for eigendecomposition
- Model order (embedding dimension)
- List of eigenmodes reconstructed
- r^2 correlation coefficient for reconstruction

Eigenvalues

This section lists the component eigenvalues, the eigenvalues normalized so that each sum to 100%, and the r^2 correlation coefficient for that component's reconstruction. Both the normalized eigenvalue %, ranging from 100 to 0, and the partial r^2 value, ranging from 1.0 to 0.0, are indicators of the measure of variance in the original data captured by that eigenmode.

Each eigenvalue and eigenvalue percent is reported, since these are part of the decomposition. Those not used in the reconstruction will be grayed. The partial r^2 values are reported only for those eigenmodes used in the reconstruction.

Estimate Menu



The **Estimate Menu** is used to estimate (interpolate) a data set using various procedures and also for AR autoregressive prediction.



The **Spline Estimation** option offers eight important spline procedures for interpolation and smoothing. First and second derivatives are also available. This procedure is useful for creating uniformly spaced data sets since uniform data are not required.



The **Fourier Estimation** option offers interpolation and smoothed estimations based upon the frequency spectrum.



The **Smoothed Data Spline Estimation** procedure combines smoothing and B-spline estimation. Note that this is not a smoothing spline, but is rather the fitting of an interpolating B-spline to data that have been pre-smoothed.



The **Local Regression Spline Estimation** option offers an adjustable order Loess-type (locally-weighted least-squares) procedure. This procedure is also useful for creating uniformly spaced data sets since uniform data are not required.



The **Savitzky-Golay Spline Estimation** procedure combines the full capability of the Savitzky-Golay smoothing with a constrained spline interpolant. This procedure has been especially tailored for estimating derivatives.



The **AR Modeling and Prediction** procedure offers effective autoregressive forecasting and extrapolation. The AR algorithms include SVD (singular value decomposition) procedures for in-place noise removal. Multiple orders can be simultaneously plotted and stabilizations are available for roots that lie outside the unit circle. The points that are to be processed can be specified, allowing predictions based on a data segment to be compared with subsequent data. The extent of the prediction is variable and white noise can be added to test the robustness of an algorithm's prediction.

Spline Estimation

The [Spline Estimation](#) option is useful for difficult to fit data and when only a simple interpolation or estimation is needed.

The interpolation algorithms pass exactly through each data point:

- [Cubic](#)
- [Cubic constrnd](#)
- [B-Spline](#)

The smoothing algorithms are:

- [Cubic smooth](#)
- [LS B-Spline fix](#)
- [LS B-Spline opt](#)
- [LS B-Spline usr](#)
- [NURBS](#)

TableCurve 2D offers a selection of some of the finer spline procedures currently in use. For more information on spline algorithms, you may wish to refer to Carl de Boor, "A Practical Guide to Splines", 1978, Springer-Verlag, New York, ISBN 0-387-90356-9. A large portion of the spline algorithms in modern use can be traced to the Fortran routines in this excellent reference.

Algorithm

The [Cubic](#) option fits a cubic spline interpolant where the breakpoints or knots of the spline are the X values of the data. As such, the knots and order are fixed. Endpoint conditions are automatically determined, these corresponding to the not-a-knot condition described by de Boor, which requires that the third derivative of the spline be continuous at the second and next-to-last breakpoint. The spline passes exactly through each data point. for more details, see de Boor's CUBSPL (pages 55–59), or the routine CSINT in the IMSL library.

The [Cubic constrnd](#) option is similar, except that a method by Akima (H. Akima, "A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures, J. ACM 17, 1970, 589-602 or de Boor, p.53 or CSAKM in the IMSL library) is used to minimize the 'wiggles' common to cubic splines. The wild swings observed when noisy data are fit to cubic splines do not occur with this algorithm. Again, the spline passes exactly through each data point.

The [Cubic smooth](#) option fits a smooth cubic spline to the data. The smoothing is automatically determined using a procedure known as cross-validation. Since the interpolant will not pass exactly through the data, TableCurve 2D reports a standard error (SE) and r^2 for this type of spline. For more information on smoothing splines, refer to deBoor's SMOOTH (p235-243), for cross validation to P. Craven and G. Wahba (1979), and to CSSCV in the IMSL library.

The [B-Spline](#) option fits the B-spline of a specified order using a knot sequence based upon the X-values of the data. The knots are thus fixed. Permissible orders range from 2 (quadratic) through order 8. The B-spline passes exactly through each data point. For more information, see de Boor's SPLINT (p204-208) or BSINT in the IMSL library.

The [LS B-Spline fix](#) option offers a B-spline where the knots are not located at the X-values, but fixed in position, and lower in count than the number of data points. For this option, you must set both the order and the number of knots desired. The smaller the number of knots, the greater will be the smoothing of noisy data. TableCurve 2D reports a standard error (SE) and r^2 for this type of spline. This algorithm

produces a least-squares minimization. For more information, refer to de Boor's L2APPR (p255-258) or to BSLSQ in the IMSL library.

The **LS B-Spline opt** option is identical to the previous spline, except that the knots are now variable, and are shifted to further minimize the overall least squares fit. Again, you must set both the order and the number of knots desired and a standard error (SE) and r^2 will be reported. Due to its iterative nature, this algorithm is appreciably slower than all of the other spline options. For more information, refer to de Boor's NEWNOT (p184-186,258-261) or to BSVLS in the IMSL library.

The **LS B-Spline usr** option offers a B-spline where the knots are now located at user-specified X-values. For this option to be available, a knot sequence must first be imported. You must also specify the spline order.

The **NURBS** option fits a non-uniform rational B-spline to the data. The knots are fixed at the X values of the data points. The order can vary from 2 to 8. Unlike the wild oscillations sometimes observed with the higher order B-splines, NURBS become smoother with increasing order. A NURBS by its nature is a smoothing spline, and does not pass exactly through the data points. A standard error (SE) and r^2 is thus reported. Because NURBS have become the model of choice for high-end 3D modeling applications, much information is available in the OpenGL materials. Another reference is W. T. Hewitt and D. Yi, "The NURBS Procedure Library", AGOCG, May 1992.

Order

The order can be set for the B-Spline options and the NURBS option. The cubic splines are obviously fixed at order 3.

Knots

The knots can only be set for the three least-squares B-spline options which allow variable knots. It is recommended that you start with a small number of knots, and increase the knot count until a satisfactory fit is obtained. the goodness of fit will improve with knot count, but this means rather little. A B-Spline with a count of fixed knots equal to the X values produces the standard B-spline, a perfect fit that passes exactly through each point.



The **Import Knot Sequence** button is used to import a user-defined knot sequence. Once a valid knot sequence has been imported, the **LS B-Spline usr** algorithm will be enabled. The knot sequence can be contained in any of TableCurve 2D's supported **import formats**. Specify only the interior knots in the sequence. The knots at the data bounds are automatically added.

Output

The principal output will interpolate the spline model or **Function**. Mainly for the procedures that perform smoothing, it is also possible to directly evaluate the first or second derivative (**1st Deriv**, **2nd Deriv**).

For derivatives, be very cautious with noisy data. A significant level of smoothing is generally needed to produce accurate derivatives. With noisy data, a spline that performs significant smoothing is essential for valid derivatives. The **Savitzky-Golay Spline Estimation** option is recommended for serious derivative evaluations.

A quadratic spline has a continuous first derivative. A cubic spline is smooth in the first derivative and continuous in the second derivative. A quartic spline is smooth in both first and second derivatives and continuous in the third derivative. If you need a smooth derivative, be sure you use a spline order two greater than the derivative desired.

For the output, the number of elements **n** are generated between **x start** and **x end**. Up to 65536 values

can be generated. The starting and ending X values should be at or within the range of the input data. The default values will perform a 2x upsampling within the x range of the data.

Sectioning Data

You may enable or disable regions of data or individual points just as within TableCurve 2D's main [Section Data](#) option. When the state of one or more points is changed, the spline is recomputed and the graph is updated.

Goodness of Fit Statistics

A standard error (SE) and r^2 are reported. These statistics are of limited use, although they do allow a comparison between a favored spline estimation and your choice of a parametric fit.

List



The [List Data](#) option lists the index, time, and output in a three column table. The listing uses the [TableCurve 2D text viewer](#) facility.

Copy



The [Copy Data to Clipboard](#) option copies the time and output values to the clipboard. Formats include full precision binary (for spreadsheets such as Excel) and ASCII (for pasting into text editors).

Save



The [Save Data to Disk](#) option writes the time and output values to a supported file format. These formats include ASCII, Excel 97/2000, Excel 95, Lotus WK3, Lotus WK1, SPSS, or Systat.

Production Facility



The [TableCurve 2D Automation](#) facility allows unattended processing of large numbers of data sets. The data sets can be consolidated in an Excel file or acquired using a DLL. The graphs can be exported to an MS Word/RTF file, while the processed data can be exported to an Excel 95 or Excel 97/2000 file.



The [Reset](#) button restores the data to its state when first entering the procedure. If an [Automation Session](#) is in progress, the [Reset](#) button can be used to terminate the automated processing.

Residuals



The [Residuals](#) button opens a [TableCurve 2D Graph](#) containing the residuals from the spline fit. The residuals are the difference between the data and spline model. This option is not available for the spline procedures that exactly interpolate the data.

Evaluation



The [Evaluation](#) option offers the means to more extensively evaluate the spline model. The evaluation can be direct, or the computation can consist of first or second derivatives, roots, or integrated areas. This option can be used to generate a table or file of any size using a generated X grid or by importing X data from supported file formats.

MS Word/RTF Export



The [MS Word/RTF File Export](#) option is used to save the current graph to either an MS Word file or a

portable RTF (Rich Text Format) file. The graph is inserted into the file as a Windows metafile.

Automatic Update

The List, Residuals, and Evaluation windows are automatically updated when any change is made to the algorithm's settings.

Updating the Data Table



When exiting this procedure with the **OK** button, an option will be presented to update TableCurve 2D's main data table with the estimated data.

If **Background Thread Fitting** is active, the fitting will be initiated as soon as the estimation is accepted and the data is rescanned.

Fourier Estimation

The **Fourier Estimation** option in the Estimate menu or the **Process toolbar** offers true interpolation and smoothing based upon the frequency spectrum. When all frequency channels are active, interpolation occurs. When thresholds are set, this option performs filtering and reconstruction. For performance reasons, this procedure is not recommended for very large data sets.

Uniformly spaced data are needed so that no additional noise is introduced. This procedure does not automatically generate uniform data for the Fourier decomposition. If needed, the **Spline Estimation** or **Local Regression Spline Estimation** option can be used to create an interpolated data set with constant X-spacing.

This option presents a dual **TableCurve 2D graph** with the **frequency domain decomposition** in the upper graph and the input (unfiltered) and output (filtered) time domain data in the lower graph.

Interpolation

Fourier interpolation is based upon the frequency spectrum. This means that any size reconstruction will generate the same Fourier spectrum. For interpolation only, the thresholds should remain empty.

Filtering

This procedure is similar to the **Fourier Filtering** option, in that frequency and spectral thresholds can be set which include or exclude specific FFT channels from the reconstruction.

Reconstruction

In this procedure, the reconstructed data are computed directly from the amplitude, frequency, and phase of the continuous sine components. While this can be appreciably slower than an inverse FFT, this does offer a reconstruction where the data count and limits are variable. In addition to reconstructing the basic data, this option also makes it possible to reconstruct the first, second, third, or fourth derivatives. If a derivative is reconstructed, the thresholding should be specified to sufficiently smooth the data in order to produce the cleanest possible derivative.

Post-Filtering

When this option is used to increase the data count within the original time interval, the Nyquist frequency will increase. A follow up FFT of the reconstructed data will contain frequencies that lie beyond those in the original FFT. This may be fine, representing a valid extrapolation and gentle decay. On the other hand, all frequencies beyond the original Nyquist can be zeroed by checking the **Low Pass** post-filter option.

Transform

This procedure fixes the FFT method to the **Best Exact N** algorithm. It is used to produce the **Fourier decomposition** and for the forward and inverse transforms in the post-filter procedure. The **Best Exact N** composite algorithm produces the fastest possible exact n FFT.

Plot

The frequency domain information can be plotted in a variety of formats. In the following table, Re is the real component of the FFT at a given frequency, Im is the imaginary component, n is the data set size, dx is the sampling interval, and var is the variance of the data series.

- **Magnitude**, $\sqrt{\text{Re} \cdot \text{Re} + \text{Im} \cdot \text{Im}}$
- **Amplitude**, $2.0 \cdot \sqrt{\text{Re} \cdot \text{Re} + \text{Im} \cdot \text{Im}} / n$
- **Phase**, sine-based, $\text{Pi}/2 + \text{atan}(\text{Im}/\text{Re})$

- **Variance**, Power normalized by variance, $(\text{Re}*\text{Re}+\text{Im}*\text{Im})/n/\text{var}$
- **dB**, decibels, $10.0*\log_{10}(\text{Re}*\text{Re}+\text{Im}*\text{Im})$
- **dB Norm**, decibels, normalized to 0 for frequency channel with maximum power
- **PSD SumSq**, Power as Sum Squared Amplitude, $2.0*(\text{Re}*\text{Re}+\text{Im}*\text{Im})/n$
- **PSD MeanSq**, Power as Mean Squared Amplitude, $2.0*(\text{Re}*\text{Re}+\text{Im}*\text{Im})/n/n$
- **PSD TimeInt**, Power as Time-Integral Squared Amplitude, $2.0*dx*(\text{Re}*\text{Re}+\text{Im}*\text{Im})/n$

In an amplitude plot, you see the actual amplitude of sine components. In a normalized decibel plot, the highest peak is at 0dB, a peak at -3dB would have half the power, and a peak at -6dB would have half the amplitude. The PSD TISA (time-integral squared amplitude power) is the actual integral under the curve defined by the square of the raw data.



If the data set size is 512 or fewer (257 or fewer points in the frequency spectrum), a bar format is automatically used. Otherwise, the points in the spectrum will be rendered directly with lines connecting the points. You can override either of these point formats using the Modify Point Format option and their respective states will be saved across sessions.

Frequency Thresholds

There are several ways to set the thresholds for Fourier filtering. The first is to define a range of frequency space. This is primarily used to isolate and reconstruct signal components. The **Freq1** and **Freq2** fields set limits in frequency. The frequency boundaries of this region can either be included (**Incl**) or excluded (**Excl**). If a frequency range is included, this means that only the frequencies that fall within this range are used in the data reconstruction. If the region is excluded, only those frequencies that fall outside the region are used for the reconstruction.

The frequencies can be entered numerically. If both fields for frequency are left blank, the full range is assumed for the **Incl** option, and no range is specified for the **Excl** option. If only a lower limit is entered, the variable range is assumed to be from this limit and greater for the **Incl** option, and all values below this limit if the **Excl** option is set. If only an upper limit is entered, the variable range is assumed to be from this limit and lower for **Incl**, and all values above this limit for **Excl**.



It is generally easier to set a frequency range graphically. When the **X Sectioning** mode is selected, the mouse is used in the upper graph to select the frequency range of interest. Frequencies are toggled off by holding the left mouse button down and sliding right and toggled back on by moving left. Individual frequencies can be toggled on and off by placing the cursor at the point (or top of a bar) and left clicking the mouse.

Spectral Thresholds

Another way to set the thresholds for Fourier filtering involves defining spectral thresholds. This is primarily used to separate signal components by power. It can also be used for noise removal, replicating the functionality in the **Fourier Denoising** option, or for signal removal, generating a signal-free background for subsequent analysis.

The **Spec1** and **Spec2** fields set the spectral limits. These can only be entered numerically. If only **Spec1** is set, all values at or above this threshold will be used in the reconstruction if **Incl** is set, and all below this threshold will be used if **Excl** is checked. If only **Spec2** is set, all values at this threshold and below will be used in the reconstruction if **Incl** is set, and all above this threshold will be used if **Excl** is set. For this option, the normalized dB values must use the minus sign.

Frequency-Spectral Thresholds

The thresholds for Fourier filtering can also consist of a frequency-spectrum region. This allows signal components to be isolated by both frequency and spectral power. This is the most effective way to isolate

and reconstruct a single signal component when some number of components are present.

The [Freq1](#) and [Freq2](#) fields set limits in frequency while the [Spec1](#) and [Spec2](#) fields set the spectral limits. The values can be entered numerically. If only a lower limit field is set, all values at or above this threshold will be used in the reconstruction if [Incl](#) is set, and all below this threshold will be used if [Excl](#) is checked. If only the upper limit is set, all values at this threshold and below will be used in the reconstruction if [Incl](#) is set, and all above this threshold will be used if [Excl](#) is set. Again, the normalized dB values must use the minus sign.



It is generally easier to set a frequency-spectrum region graphically. When the [XY Sectioning](#) mode is selected, the mouse is used in the upper graph to enclose the frequency-spectral region of interest. To include the region that is to be graphically defined, click and hold down the left mouse button at one corner of the frequency-spectral rectangle in the upper graph and drag the rectangle so that it encloses the region desired. This will automatically enter both the frequency and spectral limits and set the process state to include the FFT channels located within the region.

To exclude the region that is to be graphically defined, the same graphical procedure is followed except that the right rather than the left mouse button is used. In this case, the process state will be set to exclude the FFT channels within the region. This is useful for reconstructing data with a single component removed, such as line noise.

Note that it is possible to reconstruct only one frequency-spectral region at a time.



The [Clear](#) button is used to clear all of the threshold fields.



The [Apply](#) button can be used to immediately update the processing rather than wait for the automatic update.

Popup Information



The [Toggle Popup Information Window](#) option is used to report estimated noise and power reductions as well as a correlation coefficient between the filtered and unfiltered data.

TableCurve 2D offers a robust noise estimation procedure that may be of some value for low-frequency signals. A cubic polynomial interpolation is made for each point using the two points to the left and the two to the right (excluding the current point). The difference between the interpolated and data values is used to generate a measure of the white noise present in the data. This assumes that the data can be locally characterized by a smooth cubic interpolant. Also, the signal component(s) should exist only in the lower quarter of the Nyquist range. If a high frequency signal component is present, these estimates of noise will be invalid. The [Noise In](#) value reports the estimated white noise in the incoming data, the [Noise Out](#) value the estimated white noise for the filtered signal. The [Noise %](#) is given as the amount of estimated noise remaining after filtration.

The [TISA In](#) value reports the TISA power in the incoming data. The [TISA Out](#) value is the TISA power for the filtered signal. The [TISA %](#) is given as the amount of power remaining after filtration.

The [r-squared correlation coefficient](#) is also reported. An r^2 of 1 is a perfect correlation while a value of 0 means the filtered and unfiltered signals are completely uncorrelated.

Output Data

Since the Fourier decomposition can be expressed as continuous infinite-length phase-bearing sinusoids, it is possible to generate interpolated output for any number of values ([n](#)) and any range of times ([Time1](#) to [Time2](#)). Since the Fourier reconstruction reproduces the same data series outside the range of original data, there is no value in generating extrapolations. This procedure cannot be used for prediction.

In addition to reconstructing the sinusoidal basis functions ([Function](#)), it is also possible to reconstruct the first through fourth derivatives ([1st Deriv](#), [2nd Deriv](#), [3rd Deriv](#), [4th Deriv](#)). For derivatives, be very cautious with noisy data. A significant level of filtering is generally needed to produce accurate derivatives. In general, the [Savitzky-Golay Spline Estimation](#) option is recommended for serious derivative evaluations.

The output data stream is limited to 16384 values.

Post-Filter

If the [Low Pass](#) item is checked, an additional Fourier-based low pass filtration is performed on the output data. All frequencies greater than the Nyquist limit of the original data series will be zeroed and an inverse FFT will generate the final output. Note that the additional frequency information arising from interpolating a higher data density is often a valid extrapolation. Unlike the Fourier Upsampling procedure, this post filter is optional.

List



The [List Data](#) option lists the index, time, and output signal in a three column table. The listing uses the [TableCurve 2D text viewer](#) facility.

Copy



The [Copy Data to Clipboard](#) option copies the time and output signal values to the clipboard. Formats include full precision binary (for spreadsheets such as Excel) and ASCII (for pasting into text editors).

Save



The [Save Data to Disk](#) option writes the time and output values to a supported file format. These formats include ASCII, Excel 97/2000, Excel 95, Lotus WK3, Lotus WK1, SPSS, or Systat.

Production Facility



The [TableCurve 2D Automation facility](#) allows unattended processing of large numbers of data sets. The data sets can be consolidated in an Excel file or acquired using a DLL. The graphs and reports can be exported to an MS Word/RTF file, while the processed data can be exported to an Excel 95 or Excel 97/2000 file.



The [Reset](#) button restores the data to its state when first entering the procedure. If an [Automation Session](#) is in progress, the [Reset](#) button can be used to terminate the automated processing.

Residuals



The [Residuals](#) button is used to graphically display the difference between the original and filtered data.

Numeric Summary



The [Numeric Summary](#) offers a full FFT report. The report optionally includes a listing of the interpolated spectral peaks, a frequency analysis, and a linear sinusoidal least-squares fit summary.

FFT Summary



The [List Frequency Domain Data](#) option offers an extended FFT data summary. The listing uses the [TableCurve 2D text viewer](#) facility. The FFT channel number, frequency, and magnitude are always listed.

The **Format** menu offers the optional selection of the following:

- Add Real,Imag
- Add Amplitude
- Add Wavelength
- Add Phase (Sine-based)
- Add dB
- or Add dB Normalized
- Add Power Spectral Density, Sum Squared Amplitude
- or Add Power Spectral Density, Mean Squared Amplitude
- or Add Power Spectral Density, Time-Integral Squared Amplitude

In the following table, **Re** is the real component of the FFT at a given frequency, **Im** is the imaginary component, **n** is the data set size, and **dx** is the sampling interval:

- **Magnitude**, $\sqrt{\text{Re}*\text{Re}+\text{Im}*\text{Im}}$
- **Phase**, sine-based, $\text{Pi}/2+\text{atan}(\text{Im}/\text{Re})$
- **Amplitude**, $2.0*\sqrt{\text{Re}*\text{Re}+\text{Im}*\text{Im}}/n$
- **dB**, decibels, $10.0*\log_{10}(\text{Re}*\text{Re}+\text{Im}*\text{Im})$
- **dB Norm**, decibels, normalized to 0 for frequency channel with maximum power
- **PSD SumSq**, Power as Sum Squared Amplitude, $2.0*(\text{Re}*\text{Re}+\text{Im}*\text{Im})/n$
- **PSD MeanSq**, Power as Mean Squared Amplitude, $2.0*(\text{Re}*\text{Re}+\text{Im}*\text{Im})/n/n$
- **PSD TimeInt**, Power as Time-Integral Squared Amplitude, $2.0*dx*(\text{Re}*\text{Re}+\text{Im}*\text{Im})/n$

In an amplitude plot, you see the actual amplitude of sine components. In a normalized decibel plot, the highest peak is at 0dB, a peak at -3dB would have half the power, and a peak at -6dB would have half the amplitude. The PSD TISA (time-integral squared amplitude power) is the actual integral under the curve defined by the square of the raw data.

The amplitude and phase of each component in the FFT is derived from sine-based conversion. Each of the components in the FFT can be reconstructed using:

$$Y=\text{Amplitude}*\sin(2*\text{PI}*\text{Frequency}*X+\text{Phase})$$

MS Word/RTF Export



The **MS Word/RTF File Export** option is used to save the current graphs and reports to either an MS Word file or a portable RTF (Rich Text Format) file. The graphs are inserted into the file as Windows metafiles.

Evaluation



The **Evaluation** option offers the means to more extensively evaluate the sum of sinusoids model generated by the **Fourier decomposition**. The evaluation can be direct, or the computation can consist of first or second derivatives, roots, or integrated areas. This option can be used to generate a table or file of any size using a generated X grid or by importing X data from supported file formats.

Automatic Update

The List, Residuals, Numeric Summary, Fourier Summary, and Evaluation windows are automatically updated when any change is made to the algorithm's settings.

Updating the Data Table

 When exiting this procedure with the **OK** button, an option will be presented to update TableCurve 2D's main data table with the estimated data.

If **Background Thread Fitting** is active, the fitting will be initiated as soon as the estimation is accepted and the data is rescanned.



Smoothed Data Spline Estimation

The [Smoothed Data Spline Estimation](#) procedure combines TableCurve 2D's [Smoothing](#) and [Spline Estimation](#). Note that this is not a smoothing spline, but is rather the fitting of an interpolating B-spline to data that have been pre-smoothed.

Unless the Loess algorithm is used, uniformly spaced data are needed so that no additional noise is introduced. This procedure does not automatically generate uniform data for the smoothing algorithms. If needed, the [Spline Estimation](#) or [Local Regression Spline Estimation](#) option can be used to create an interpolated data set with constant X-spacing.

Smoothing Level, AI Expert, and Smoothing Algorithm

These are discussed in the [Automated Smoothing](#) option.

B-Spline Order

This is discussed in the [Spline Estimation](#) option. Note that the B-Spline will exactly pass through the smoothed data.

Goodness of Fit Statistics

A standard error (SE) and r^2 are reported. This is based upon the raw data and final spline interpolant. These statistics are of limited use, since a zero smoothing level and B-spline produce a perfect fit of noisy data. They may, however, allow a comparison between a favorable estimation and your choice of a parametric fit.

Output

The principal output will interpolate the spline model or [Function](#). It is also possible to directly evaluate the first or second derivative ([1st Deriv](#), [2nd Deriv](#)).

For derivatives, be very cautious with noisy data. A significant level of smoothing is generally needed to produce accurate derivatives. With noisy data, significant smoothing is essential for valid derivatives. The separate [Savitzky-Golay Spline Estimation](#) option is recommended for serious derivative evaluations.

A quadratic spline has a continuous first derivative. A cubic spline is smooth in the first derivative and continuous in the second derivative. A quartic spline is smooth in both first and second derivatives and continuous in the third derivative. If you need a smooth derivative, be sure you use a spline order two greater than the derivative desired.

For the output, the number of elements n are generated between [x start](#) and [x end](#). Up to 65536 values can be generated. The starting and ending X values should be at or within the range of the input data. The default values will perform a 2x upsampling within the x range of the data.

List



The [List Data](#) option lists the index, time, and output in a three column table. The listing uses the [TableCurve 2D text viewer](#) facility.

Copy



The [Copy Data to Clipboard](#) option copies the time and output values to the clipboard. Formats include full precision binary (for spreadsheets such as Excel) and ASCII (for pasting into text editors).

Save



The **Save Data to Disk** option writes the time and output values to a supported file format. These formats include ASCII, Excel 97/2000, Excel 95, Lotus WK3, Lotus WK1, SPSS, or Systat.

Production Facility



The **TableCurve 2D Automation** facility allows unattended processing of large numbers of data sets. The data sets can be consolidated in an Excel file or acquired using a DLL. The graphs can be exported to an MS Word/RTF file, while the processed data can be exported to an Excel 95 or Excel 97/2000 file.



The **Reset** button restores the data to its state when first entering the procedure. If an **Automation Session** is in progress, the **Reset** button can be used to terminate the automated processing.

Residuals



The **Residuals** button opens a **TableCurve 2D Graph** containing the residuals from the spline fit. The residuals are the difference between the data and spline model. This option is not available for the spline procedures that exactly interpolate the data.

Evaluation



The **Evaluation** option offers the means to more extensively evaluate the spline model. The evaluation can be direct, or the computation can consist of first or second derivatives, roots, or integrated areas. This option can be used to generate a table or file of any size using a generated X grid or by importing X data from supported file formats.

MS Word/RTF Export



The **MS Word/RTF File Export** option is used to save the current graphs to either an MS Word file or a portable RTF (Rich Text Format) file. The graph is inserted into the file as a Windows metafile.

Automatic Update

The List, Residuals, and Evaluation windows are automatically updated when any change is made to the algorithm's settings.

Updating the Data Table



When exiting this procedure with the **OK** button, an option will be presented to update TableCurve 2D's main data table with the estimated data.

If **Background Thread Fitting** is active, the fitting will be initiated as soon as the estimation is accepted and the data is rescanned.

Local Regression Spline Estimation

The **Local Regression Spline Estimation** option in the Estimate menu or the **Process toolbar** offers an adjustable order Loess-type (locally-weighted least-squares) procedure with two different weighting functions. This procedure combines Loess fitting, a discontinuous smoothing method, with the smoothness and continuity of a B-spline. This procedure can sometimes extract an underlying low frequency data pattern in extremely noisy data. This procedure is also useful for both upsampling and downsampling since the range and number of output points are specified. Uniform data are not required.

This option can be extremely slow and is not recommended for very large data sets.

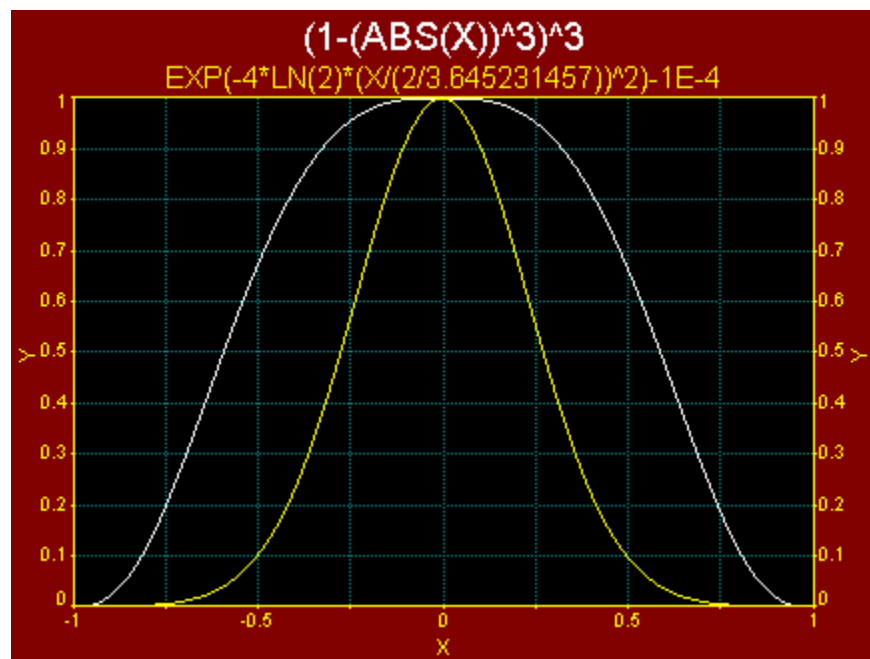
The input data and local regression curve are shown in a **TableCurve 2D Graph** for rapid visualization of the effectiveness of the fit.

Algorithm

This procedure is a fitting and smoothing algorithm that performs a series of weighted least squares fits in a moving window across the data. The TableCurve 2D local regression estimation algorithm is similar to the Loess non-parametric estimation procedure. The TableCurve 2D algorithm also offers a Gaussian weighting function in addition to the tricube weighting function used in Loess. It also offers variable order fitting, from linear through quartic. For more information on the Loess algorithm, you may wish to refer to William S. Cleveland, "Visualizing Data", 1993, Hobart Press, Summit, NJ, ISBN 0-9634884-0-6.

Weighting Function

The tricube weighting function gives a much greater weight to nearby points as compared to a Gaussian. The following graph depicts both the tricube function and the particular form of Gaussian TableCurve 2D implements for this algorithm:



Model

You can choose to locally fit a linear (order 1), quadratic (order 2), cubic (order 3), or quartic (order 4) model to the data.

Data Window

The window minimum value incurs the minimum of smoothing. Higher window counts can be used to increasingly smooth the data.

Sectioning Data

You may enable or disable regions of data or individual points just as within TableCurve 2D's main [Section Data](#) option. When the state of one or more points is changed, the local regression and B-spline interpolant is recomputed and the graph is updated.

B-Spline Order

This is discussed in the [Spline Estimation](#) option. Note that the B-Spline will exactly pass through the smoothed data.

Goodness of Fit Statistics

A standard error (SE) and r^2 are reported. This is based upon the raw data and final spline interpolant. These statistics are of limited use, since a zero smoothing level and B-spline produce a perfect fit of noisy data. They may, however, allow a comparison between a favorable estimation and your choice of a parametric fit.

Output

The principal output will interpolate the spline model or [Function](#). It is also possible to directly evaluate the first or second derivative ([1st Deriv](#), [2nd Deriv](#)).

For derivatives, be very cautious with noisy data. A significant level of smoothing is generally needed to produce accurate derivatives. With noisy data, significant smoothing is essential for valid derivatives. The separate [Savitzky-Golay Spline Estimation](#) option is recommended for serious derivative evaluations.

A quadratic spline has a continuous first derivative. A cubic spline is smooth in the first derivative and continuous in the second derivative. A quartic spline is smooth in both first and second derivatives and continuous in the third derivative. If you need a smooth derivative, be sure you use a spline order two greater than the derivative desired.

For the output, the number of elements n are generated between [x start](#) and [x end](#). Up to 65536 values can be generated. The starting and ending X values should be at or within the range of the input data. The default values will perform a 2x upsampling within the x range of the data.

List



The [List Data](#) option lists the index, time, and output in a three column table. The listing uses the [TableCurve 2D text viewer](#) facility.

Copy



The [Copy Data to Clipboard](#) option copies the time and output values to the clipboard. Formats include full precision binary (for spreadsheets such as Excel) and ASCII (for pasting into text editors).

Save



The [Save Data to Disk](#) option writes the time and output values to a supported file format. These formats include ASCII, Excel 97/2000, Excel 95, Lotus WK3, Lotus WK1, SPSS, or Systat.

Production Facility



The **TableCurve 2D Automation** facility allows unattended processing of large numbers of data sets. The data sets can be consolidated in an Excel file or acquired using a DLL. The graphs can be exported to an MS Word/RTF file, while the processed data can be exported to an Excel 95 or Excel 97/2000 file.



The **Reset** button restores the data to its state when first entering the procedure. If an **Automation Session** is in progress, the **Reset** button can be used to terminate the automated processing.

Residuals



The **Residuals** button opens a **TableCurve 2D Graph** containing the residuals from the spline fit. The residuals are the difference between the data and spline model. This option is not available for the spline procedures that exactly interpolate the data.

Evaluation



The **Evaluation** option offers the means to more extensively evaluate the spline model. The evaluation can be direct, or the computation can consist of first or second derivatives, roots, or integrated areas. This option can be used to generate a table or file of any size using a generated X grid or by importing X data from supported file formats.

MS Word/RTF Export



The **MS Word/RTF File Export** option is used to save the current graphs to either an MS Word file or a portable RTF (Rich Text Format) file. The graph is inserted into the file as a Windows metafile.

Automatic Update

The List, Residuals, and Evaluation windows are automatically updated when any change is made to the algorithm's settings.

Updating the Data Table



When exiting this procedure with the **OK** button, an option will be presented to update TableCurve 2D's main data table with the estimated data.

If **Background Thread Fitting** is active, the fitting will be initiated as soon as the estimation is accepted and the data is rescanned.



Savitzky-Golay Spline Estimation

The [Savitzky-Golay Spline Estimation](#) procedure combines the full capability of the [Savitzky-Golay Smoothing](#) option with a [B-Spline Interpolant](#).

Like the [Savitzky-Golay Smoothing](#) option, this procedure directly generates a filter for the various derivative orders. The Savitzky-Golay algorithm is generally the best way to produce accurate smooth derivatives.

This procedure does not automatically generate the uniformly-spaced data required by the Savitzky-Golay algorithm. If needed, the [Spline Estimation](#) or [Local Regression Spline Estimation](#) option can be used to create an interpolated data set with constant X-spacing.

Win n, Order, Passes and Zero Edges

These Savitzky-Golay algorithm controls are discussed in the [Savitzky-Golay Smoothing](#) option

Constrained Cubic Spline

This is discussed in the [Spline Estimation](#) option. The constrained cubic spline interpolant, which has no adjustments, will exactly pass through the smoothed data but will not evidence the wilder oscillations common to splines. This algorithm was selected to further stabilize estimates of derivatives, particularly those beyond first order.

Output

The principal output will interpolate the results from the zero-order Savitzky-Golay filter or [Function](#). When the first, second, third, or fourth derivative ([1st Deriv](#), [2nd Deriv](#), [3rd Deriv](#), [4th Deriv](#)) is selected, the constrained cubic spline interpolates the results from the specified derivative filter.

Note that the derivatives are based on the Savitzky-Golay filter, not the interpolating spline as is the case in the [Smoothed Data Spline Estimation](#) procedure. In that instance, smoothness and continuity of the derivative depended on the order of the spline. In this option, all derivative estimations are continuous and smooth, regardless of the derivative order being estimated.

For the output, the number of elements [n](#) are generated between [x start](#) and [x end](#). Up to 65536 values can be generated. The starting and ending X values should be at or within the range of the input data. The default values will perform a 2x upsampling within the x range of the data.

Estimated Noise Reduction

TableCurve 2D offers a robust noise estimation procedure that may be of some value for low-frequency signals. A cubic polynomial interpolation is made for each point using the two points to the left and the two to the right (excluding the current point). The difference between the interpolated and data values is used to generate a measure of the white noise present. This assumes that the data can be locally characterized by a smooth cubic interpolant.

The [Raw](#) value reports the estimated white noise in the incoming data, the [Smooth](#) value the estimated white noise for the smoothed data. The percent is given as the amount of estimated noise remaining after smoothing.

The [r-squared correlation coefficient](#) is also reported. An r^2 of 1 is a perfect correlation while a value of 0 means the smoothed and unsmoothed data are completely uncorrelated.

List



The **List Data** option lists the index, time, and output in a three column table. The listing uses the **TableCurve 2D text viewer** facility.

Copy



The **Copy Data to Clipboard** option copies the time and output values to the clipboard. Formats include full precision binary (for spreadsheets such as Excel) and ASCII (for pasting into text editors).

Save



The **Save Data to Disk** option writes the time and output values to a supported file format. These formats include ASCII, Excel 97/2000, Excel 95, Lotus WK3, Lotus WK1, SPSS, or Systat.

Production Facility



The **TableCurve 2D Automation** facility allows unattended processing of large numbers of data sets. The data sets can be consolidated in an Excel file or acquired using a DLL. The graphs and reports can be exported to an MS Word/RTF file, while the processed data can be exported to an Excel 95 or Excel 97/2000 file.



The **Reset** button restores the data to its state when first entering the procedure. If an **Automation Session** is in progress, the **Reset** button can be used to terminate the automated processing.

Residuals



The **Residuals** button is used to graphically display the difference between the original and smoothed data.

Numeric Summary



The **Numeric Summary** displays a basic summary of the analysis and a listing of the Savitzky-Golay filter coefficients. If a derivative is currently selected, both the zero order data and the specified derivative coefficients are listed.

Evaluation



The **Evaluation** option offers the means to more extensively evaluate the spline model. The evaluation can be direct, or the computation can consist of first or second derivatives, roots, or integrated areas. This option can be used to generate a table or file of any size using a generated X grid or by importing X data from supported file formats.

MS Word/RTF Export



The **MS Word/RTF File Export** option is used to save the current graphs and reports to either a MS Word file or a portable RTF (Rich Text Format) file. The graphs are inserted into the file as Windows metafiles.

Automatic Update

The List, Residuals, Numeric Summary, and Evaluation windows are automatically updated when any change is made to the algorithm's settings.

Updating the Data Table



When exiting this procedure with the **OK** button, an option will be presented to update TableCurve 2D's main data table with the estimated data.

If **Background Thread Fitting** is active, the fitting will be initiated as soon as the estimation is accepted and the data is rescanned.

AR Modeling and Prediction

The **AR Modeling and Prediction** procedure in the **Estimate** menu or the **Process** toolbar offers effective forecasting and extrapolation using **autoregressive modeling**. The AR model is either forward (subsequent points are predicted), backward (prior points are predicted), or forward-backward (both subsequent and prior points are predicted). The algorithms include SVD (singular value decomposition) methods for in-situ (in place) noise removal. The two standard stabilizations are available for roots that lie outside the unit circle.

Uniformly-Spaced Data

This procedure does not automatically generate the uniformly-spaced data required by the autoregressive algorithms. If needed, the **Spline Estimation** or **Local Regression Spline Estimation** option can be used to create an interpolated data set with constant X-spacing.

Discrete Output

An autoregressive model produces discrete output; it does not offer interpolation. One of the key uses for AR modeling is to generate a series of extrapolated points that can then be used for extending a parametric fit. The parametric fit of the original data may not offer a valid extrapolation, but a parametric fit of data containing these predicted points may well do so.

Multiple Orders

Multiple AR model order predictions can be simultaneously plotted and an average curve can be used for the prediction. **Error bars** simplify the task of assessing the impact of model order on the predicted values.

Assessing the Validity of Predictions

The points that are to be processed can be specified, allowing predictions based on a data segment to be compared with actual subsequent data. The extent of the prediction is variable and white noise can be added to test the algorithm's robustness.

Graph

The input data and the AR model fit and prediction for each selected order are plotted in a **TableCurve 2D Graph**.



The individual predictions and their labels can be toggled on and off with the added reference buttons in the **graph's toolbar**.



The **graph's toolbar** also has a button that offers the full selection of error bars, as well as an **error bar** toggle.

Algorithm

The **AR algorithm list** offers nine effective AR least-squares procedures. The initial three listed are forward prediction algorithms (**Fwd**). These are the most commonly used algorithms since they are used to predict future values. The next three are backward prediction algorithms (**Bwd**). These are used to predict the data existing prior to the start of a data sequence. The final three are forward-backward prediction algorithms (**FB**) which predict both prior and subsequent time series values.

You must select a **Fwd** algorithm to predict future values, a **Bwd** algorithm to predict past values, and an

FB algorithm to predict in both directions.

The **Data** algorithms process a full forward, backward, or forward-backward prediction data matrix. This is the most accurate matrix processing procedure. The **Nrml** algorithms process the normal equations, a square matrix that can offer significant performance benefits with very large data sets. An AR model fit is a linear non-iterative matrix method, and the **Data** method should offer sufficient performance except perhaps for the very largest data sets. Typically, the **Data** algorithms, both with and without **SVD**, produce stable estimates where all roots lie on or within the unit circle. This is not assured, though.

The **SVD** algorithms offer in-situ noise removal. With an effective signal-noise threshold selection, the **SVD** procedures will most certainly offer the most accurate predictions. The predictions are based only on the signal components in the data and are minimally impacted by noise. **SVD** procedures are up to an order of magnitude slower than their non-SVD counterparts. Apart from processing time, there are few drawbacks to using **SVD** and a great many benefits.

To be consistent, the statistical results reported for all AR fits are based on a forward prediction filter with partial backward prediction. This will not exactly match any of the linear AR procedures. The **AR filter** is defined using backward prediction from the model order down to the initial data element, and forward prediction from the model order up to the final data element. This simplifies the goodness of fit statistics since a single estimate is made for each of the input data elements. These statistics should be generally valid for the Fwd models, but they should be used very cautiously as a basis for selecting Bwd prediction algorithms or orders.

Model Order Selection

If the **Single Order** option is checked, the **order** field is used to enter the desired AR model order. If this box is not checked, the order minimum (**min**), maximum (**max**), and increment (**inc**) are specified. The **Data Fwd**, **Data Bwd**, and **Data FB** least-squares methods compute all lesser orders when computing a given order, so these algorithms will be very fast in this procedure, regardless of the count of orders plotted. The **Nrml** algorithms and all of the **SVD** procedures compute only the target order, so the number of individual orders specified can significantly impact processing time. When multiple orders are fitted, an average is computed, plotted, and used for output of the predicted values.



Since the **Data** least-squares methods compute all lesser orders in the process of computing the target order, the **Plot Selection Criteria** option will include all orders up the maximum order specified. Since the **Nrml** algorithms and all of the **SVD** procedures compute only the evaluated order, the **Plot Selection Criteria** option will include only those orders actually specified by the minimum, maximum, and increment values.

When data consist of harmonic signals in the absence of noise, the minimum order needed will be twice the number of sinusoidal components. An AR model can have both real and complex roots. The real roots, usually at -0.5, 0, or 0.5 normalized frequencies, are not processed since they represent singularities at the bounds. The complex roots produce finite spectral power, and for real data, the positive and negative frequency roots mirror one another. Although TableCurve 2D reports only the positive roots, both sides of the spectrum must be taken into account. This is why the minimum order needed must be twice the number of component sinusoids.

In practice, there is usually some level of noise present in the data and a higher order model is needed. It is often the case that the minimum order needed to resolve all components is significantly higher. You may want to be generous when setting the maximum order value if it is your aim to determine this minimum order needed. Also, signals are often composed of near-harmonic and anharmonic oscillations which require higher model orders.

To achieve a reasonable signal-noise separation with **SVD**, it is necessary to fit a high enough order so that the primary singular vectors (eigenvectors) span only signal space. Provided the order is sufficiently high to produce an effective partitioning of the signal and noise, the actual order of the fit is not critical.

The quality of the fit for the noise components is not a consideration, since these eigenvectors are discarded in the SVD processing. All that is needed is an accurate specification of the **signal space**.

Signal Subspace Selection



The **Graphically Select Signal and Noise Sub-Spaces** option is available only when an **SVD** algorithm is being used. Even when a harmonic component count is known, you should use this option to insure that a high enough order is being used to achieve the desired signal-noise separation. The **eigendecomposition** for only the first of the orders fitted (the minimum order) is available for graphical signal space selection. Again, to accommodate both positive and negative frequencies, the signal space value must be twice the number of components.

When there is sufficient **signal-noise separation** in the eigenmodes, the **singular value plot** reveals one or more sharp transitions between the signal subspace and the noise subspace floor. The last eigenmode before the long sloping noise floor represents the last element of signal space. Assuming a high-enough AR model order is used, this signal-noise space separation does not become difficult until the noise level approaches that of the signal. At this point, the sharp characteristic transition disappears. An earlier diminishing of this transition occurs when the noise is red.

A full signal space SVD fit, one where the signal space equals the model order, produces the same results as the non-SVD algorithms.

Stabilization

A stable AR filter is one with roots that lie on or within the unit circle. While the least-squares AR procedures often produce fully stable coefficients, this is not assured.



The roots of the AR model can be inspected with the **Plot Roots** option. Roots consisting of signal will rest on or close to the unit circle while those corresponding with noise tend to be found in the interior of the unit circle.

In those cases where roots are outside the unit circle, they may be attributable to either signal or noise. The **Reflect Out** option will reflect the outlier root across the unit circle into the interior. Those very close to the unit circle will remain so, and will generally be processed as signal. Those well away from the unit circle will be reflected significantly into the interior and will be processed as noise. The **Unit Circle Out** will reflect all outlier roots to the unit circle, assuming that they represent valid sinusoidal components.

Data Processed

To facilitate prediction tests, it is possible to specify that only a portion of the data set be processed. The **x start** and **x end** values specify the starting and ending x values within the data stream that will be used to create the parametric model. Subsequent (or earlier) data can then be compared with the predicted values. The default values will specify the actual range of the input data.

Note that the order will be reduced automatically when it is required by the size of the subset being processed.

Predicted Points

For the output, the number of predicted elements **n** is specified. These will be appended to the end of the output data stream if a **Fwd** algorithm is used, to the beginning of the output data stream if a **Bwd** algorithm is selected, or to both ends of the data stream for an **FB** algorithm. By default, **n** will be one-quarter the number of data elements.

The output data stream will consist of the predicted AR model values for all x-values in the data series and for this number of predicted values following, preceding, or both following and preceding. If multiple

orders are used, these values will consist of an arithmetic average of all orders specified.

Although the n predicted points will use exclusively forward or backward prediction, the output data stream values corresponding with the x -values in the input data will be based upon an AR filter that is defined using backward prediction from the model order down to the initial data element, and forward prediction from the model order up to the final data element.

Add Noise

It may be instructive to see where a given procedure starts to break down as a consequence of temporarily adding white observation noise to the input data. The zero noise level is $S/N=300\text{dB}$ (fractional noise= $1\text{E-}15$, the IEEE double precision threshold for addition). At this value, no noise is added to the data. A value of 280 would add noise in the 14th significant figure, 260 in the 13th, 240 in the 12th and so on. This option assumes that the current data set is entirely signal, and adds noise accordingly. Typical test values are 40dB(1% noise), 20(10%), 10(31.6%), 6(50.1%), 3(70.8%), and 0(100%).

The **SVD** procedures have the greatest noise resistance. This noise option is also helpful in ascertaining at what level the SVD procedures can no longer evidence the eigenmode signal to noise transition.

List



The **List Data** option lists the index, time, and output signal in a three column table. The listing uses the TableCurve 2D text viewer facility.

Copy



The **Copy Data to Clipboard** option copies the time and output signal values to the clipboard. Formats include full precision binary (for spreadsheets such as Excel) and ASCII (for pasting into text editors).

Save



The **Save Data to Disk** option writes the time and output values to a supported file format. These formats include ASCII, Excel 97/2000, Excel 95, Lotus WK3, Lotus WK1, SPSS, or Systat.

Production Facility



The **TableCurve 2D Automation** facility allows unattended processing of large numbers of data sets. The data sets can be consolidated in an Excel file or acquired using a DLL. The graphs can be exported to an MS Word/RTF file, while the processed data can be exported to an Excel 95 or Excel 97/2000 file.



The **Reset** button restores the data to its state when first entering the procedure. If an **Automation Session** is in progress, the **Reset** button can be used to terminate the automated processing.



Residuals

The **Residuals** button opens a **TableCurve 2D Graph** containing the residuals from the fit. The residuals are the difference between the data and AR model fit. A good model should produce normally distributed residuals. The **SNP** plot is particularly useful for assessing if the errors in the fit have a Gaussian distribution. Note that if multiple orders are processed, the residuals for only the first of the orders fitted, the minimum order, are displayed.

Plot Selection Criteria



The **Plot Selection Criteria** option will include all orders up to the maximum order specified for the **Data** least-squares methods. These algorithms compute all lesser orders in the process of computing the target order. The **MDL** (minimum description length) is the most widely accepted AR order selection

criterion. This option will be grayed for the all of the [SVD](#) procedures since these compute only the target order. The [MDL](#) for the current model order is displayed in the dialog.

Plot Roots



The complex roots of the AR model can be inspected with the [Plot Roots](#) option. Roots consisting of signal will rest on or close to the unit circle while those corresponding with noise tend to be found in the interior of the unit circle.

MS Word/RTF Export



The [MS Word/RTF File Export](#) option is used to save the current graphs and reports to either a MS Word file or a portable RTF (Rich Text Format) file. The graphs are inserted into the file as Windows metafiles.

Numeric Summary



The [Numeric Summary](#) displays a basic summary of the analysis and optionally the predicted values, the AR coefficients, the fit statistics, and singular values. The content is set using the Numeric Summary's [Format](#) menu.

The Numeric Summary's File menu has a [Save Predicted Values as XY File](#) option to save only the forecast points to file. There is also a [Save Coefficients as Filter File](#) option which allows filters to be saved to disk as later imported as algorithms to be applied to a different data set.

Language Code



The [Code Generation](#) option is used to write the filter coefficients to disk in C, FORTRAN, Pascal, or BASIC program code. Since an AR filter must process a discrete data stream, the current data are also saved to the same location. The data file is written with an SER (Series) extension and in an ASCII format. The first line contains the number of data values in the series and the data follow. The code for a filter is necessarily more complicated than for a parametric model where no discrete data are needed.

Import Filters



The [Import Filter from Disk](#) is used to import filters that have been previously saved for processing the current data set. You may simultaneously select as many filter (FLT) files as you like. After import, all of the filters will be available in the algorithm list. They are listed by file name, so it is important to save filters with meaningful file names. Imported filters are always treated as both forward and backward predictors, regardless of how they were originally generated.

Save Filter to Disk



The [Save Filter to Disk](#) option saves the current filter coefficients to a filter (FLT) file. This is a binary file that can only be created within TableCurve 2D.

r² Value

Because the AR spectrum is a fitting procedure, the [r² correlation coefficient](#), [F-statistic](#), and [SE](#) (standard error) are shown in informational fields. An r² of 1 represents a perfect fit whereas an r² of 0 represents no fit at all. A high r² (0.95+) is needed for good prediction accuracy. The r² value is based on a forward prediction filter with partial backward prediction. This AR filter is defined using backward prediction from the model order down to the initial data element, and forward prediction from the model order up to the final data element. Although this simplifies the goodness of fit statistics (a single estimate is made for

each of the input data elements), the value will be less meaningful for backward predictions.

Automatic Update

The List, Selection Criteria, Complex Roots, Residuals, and Numeric Summary windows are automatically updated when any change is made to the algorithm's settings.

Updating the Data Table



When exiting this procedure with the **OK** button, an option will be presented to update TableCurve 2D's main data table with the output data. If multiple orders were processed, this will be the average.

If **Background Thread Fitting** is active, the fitting will be initiated as soon as the predicted data are accepted and rescanned.

Autoregressive Modeling

AR estimation algorithms are used in the [AR Modeling and Prediction](#) option.

Autoregression

In an AR model, a value at time t is based upon a linear combination of prior values (forward prediction), upon a combination of subsequent values (backward prediction), or both (forward-backward prediction). The linear models give rise to rapid and robust computations.

AR Definitions

To preserve the degrees of freedom for statistical tests and to furnish a common reference for all AR algorithms, TableCurve 2D defines an AR model as follows:

$$y_k = \sum_{j=1}^p a_j x_{k+j} \quad k=p..1$$

$$y_k = \sum_{j=1}^p a_j x_{k-j} \quad k=p+1..N$$

In these equations x is the data series of length N and a is the autoregressive parameter array of order p . TableCurve 2D uses the positive sign (linear prediction) convention for the AR coefficients. The model is defined as reverse prediction for the first p values, and forward prediction for the remaining $N-p$ values. This definition is used for all AR fit statistics, although it is not the model fitted in any of the AR linear least-squares procedures.

Classes of AR Algorithms

The AR coefficients can be computed in a variety of ways. The coefficients can be computed from autocorrelation estimates, from partial autocorrelation (reflection) coefficients, and from least-squares matrix procedures. Further, an AR model using the autocorrelation method will depend on the truncation threshold (maximum lag) used to compute the correlations. The partial autocorrelation method will depend on the specific definition for the reflection coefficient. The least-squares methods will also yield results that are a function of how data are treated at the bounds (matrix size) as well as whether the data matrix or normal equations are fitted.

The AR algorithms in TableCurve 2D are least-squares procedures since these produce the best fits. Least-squares methods that offer in-situ separation of signal and noise through singular value decomposition (SVD) are the most robust of TableCurve 2D's AR methods. These algorithms are built into the AR procedures (there is, for example, no separate Principal Component AutoRegressive or PCAR option).

AR Prediction Models

A continuous function of time needs only the computed parameters of the model to produce an estimate for any value of time desired. Whether the original time sampling is uniform or irregular is not a factor once the model is generated. This is not true for an autoregressive model. An AR linear prediction model is a discrete function that requires uniformly sampled data.

When an AR model is evaluated in the time domain for a future predicted value, the last portion of the data sequence is filtered by the AR coefficients to generate a new element. Although an AR linear prediction model does not explicitly contain a noise component, this is intrinsically a part of the data sequence. White noise is thus managed, provided it is uncorrelated, point to point.

AR Model Limitations

In practice, it may not be possible to predict a present value from past values. The process may be stochastic (truly random) rather than deterministic. There may be no correlation between past values and the current one.

Nor is the background noise necessarily white and uncorrelated. Geophysical processes, for example, are often characterized by red noise backgrounds. For red noise, the variance decreases with increasing frequency. In some cases, the overall noise trend can be approximated by a first order AR model.

As an AR model order increases, more of the trends in the data series are incorporated. When noise is completely absent, pure harmonics are captured with an order equal to twice the number of components. More commonly, though, some level of noise is present.

Further, the narrowband signals are often anharmonic. The signal components will be close to harmonic, evidencing regular oscillations, but cannot be described by a single sinusoid or exponentially-damped sinusoid. The information necessary to describe the oscillatory trend may require that the AR model capture several cycles of the oscillation. For slowly changing patterns, this can mean appreciable model orders.

AR Signal-Noise Separation

A basic AR model fit does not offer effective signal-noise partitioning. Even if pure sinusoids are embedded in modest levels of noise, it may require an order well beyond twice the signal component count to successfully capture and spectrally render the sinusoids. In other words, if a model order is too low, only a portion of the deterministic signal is captured. The remainder is treated as part of the white noise. Spectral components are thus missed.

On the other hand, if a model order is too high, the full deterministic signal is captured but some measure of the noise is also modeled. Spurious spectral peaks can result.

There are three ways to manage this limitation. First, the noise can be filtered or removed prior to analysis. Second, an optimum AR order can be selected that captures all of the deterministic signal elements and includes as little noise as possible. The third option consists of an in-situ noise removal within the least-squares procedures that generate the coefficients.

Model Order Determination



TableCurve 2D includes an AR Order Selection Criteria option to view normalized graphs of the MDL (Minimum Descriptive Length) and AIC (Akaike Information Criterion), the two indices most commonly used to determine optimum AR order. The minima in these criteria are the recommended model orders. These indices reflect major thresholds where increasing model order results in significant improvements in the AR fit. In general, though, these criteria are of limited use. Inappropriate model orders are sometimes suggested.

Signal Eigenmode Selection

Rather than struggle to find an optimum model order, an explicit signal-noise separation can be made. As the AR order increases, there are more coefficients to map both data trends and noise. If a matrix procedure based upon principal components is then used, the initial eigenvectors will describe only the signal components. The latter eigenvectors will capture the noise contributions. Model order is then less important, since the noise vectors are discarded prior to the computation of the AR coefficients.



This in-situ signal-noise separation is accomplished using singular value decomposition (SVD) in any of the least-squares AR models. The **Graphically Select Signal and Noise Subspaces** button is found in the **AR Modeling and Prediction** procedure. A visual indication of the signal-noise threshold is often given as the last major transition prior to a long sloping noise floor. Eigenmodes that capture an oscillatory signal component usually occur in close-to-equal magnitude pairs followed by a descending transition.

In the case of white noise, this type of visual eigenspace signal-noise separation is often successful even to levels where the noise power approaches that of the signal. The S/N range where this visual transition is apparent is not as great, however, in the case of red noise.

Complex Root Inspection



There is merit in exploring the complex roots of the AR model coefficients. The **Plot Roots** option will display the poles in either a unit circle or magnitude plot. Roots closest to the unit circle are usually indicative of signal, and those more to the interior are generally due to noise. This is another tool to assist in signal component determination.

AR Algorithms

Autoregressive (AR) estimation algorithms are used in the [AR Modeling and Prediction](#) option.

Least-Squares Algorithms

The most effective AR methods involve simultaneous least-squares estimates of all AR coefficients in the model.

The [Data](#) models implement the "covariance" or "modified covariance" methods of linear prediction. The covariance in this context has nothing to do with fitting a covariance matrix. Rather, the full data or trajectory matrix, usually rectangular, is fitted directly to the vector of data elements. This approach incurs no loss of precision, but can involve lengthy fits with large data sets.

The [Nrml](#) models implement the normal equations used in typical least-squares estimations. Square matrices are formed using sums of forward or backward data elements. The sums incur a loss of precision that can result in reduced resolution (compared to direct data methods) and lessened stability of the coefficients. When some noise is present, however, the normal equations generally yield viable results.

The [Fwd](#), [Bwd](#), and [FB](#) are the forward prediction, backward prediction, and forward-backward prediction variants of the algorithms.

For both the [Data](#) and [Nrml](#) models, there is an [SVD](#) version. Apart from longer processing times, there are no disadvantages to using an SVD procedure, and the advantages are numerous when removing the influence of noise is critical to the modeling. A full signal space SVD fit, one where the signal space equals the model order, produces the same results as the non-SVD algorithms.



The [Graphically Select Signal and Noise Subspaces](#) option is available only to [SVD](#) procedures.

Non-SVD Algorithms

The [Data Fwd](#), [Data Bwd](#), and [Data FB](#) options use fast algorithms that exploit the symmetry of the data matrix. These algorithms will produce optimum least-squares AR fits without the precision losses characteristic of the normal equations approach.



Since these algorithms also compute the coefficients for all lesser model orders, [AR Model Order Selection](#) is available.

SVD Based Least-Squares Algorithms

Least-squares AR methods that offer in-situ separation of signal and noise through [SVD](#) are the most robust of all methods.

The [Data Svd FB](#) is an excellent algorithm. The forward prediction version is the [Data Svd Fwd](#). The backward prediction version is the [Data Svd Bwd](#). The main drawback of the [Data Svd](#) procedures is that they can be quite slow with large data sets.

If there is noise present (which is why SVD is used in the first place), the normal equations equivalents [Nrml Svd FB](#), [Nrml Svd Fwd](#), [Nrml Svd Bwd](#), will produce similar results far more swiftly.

The SVD algorithms compute the coefficients only for the order specified.

Least-squares AR methods that offer in-situ separation of signal and noise through [SVD](#) are the most robust of AR methods.

Filter

When TableCurve 2D computes the statistics for any AR model fit, a [consistent filter](#) is used. This enables all algorithms to be compared using a common reference. The filter is not strictly forward, backward, or forward-backward prediction. Instead it is a forward prediction with a partial backward component. The AR filter is defined using backward prediction from the model order down to the initial data element, and forward prediction from the model order up to the final data element. Using this filter, a single estimate is thus made for each of the input data elements. There is no gap at one end of the data stream as is true of a forward or backward filter.

This filter simplifies the goodness-of-fit statistics since the degree of freedom is preserved (the data length minus the AR coefficient count). The statistics in the AR algorithms reflect this approach, although none of the linear AR algorithms specifically optimizes this particular merit function.

AR Model Order Selection

This option is available when AR fit information is available at multiple orders. In the following descriptions, **n** is the data length, **SSE** is normalized error in the AR model fit, **order** is the AR model order, and **SSM** is the sum of squares about the mean for the original data.

Selection Criteria

The **AIC** (Akaike Information Criterion) is defined as $n \cdot \log(\text{SSE}) + \text{order} \cdot 2$.

The **MDL** (Minimum Description Length) is defined as $n \cdot \log(\text{SSE}) + \text{order} \cdot \log(n)$.

The criteria are graphed using normalizations where the values are mapped from 1E-4 to 1.0. This is a convenience for comparing the different criteria. The minimum in the curve is supposedly the optimum AR model order. On the whole, the **MDL** is generally regarded as the best overall AR order selection criterion.

In practice, the use of any order selection criterion is fraught with considerable peril, especially when applied blindly. If you are using the **non-SVD Data** options, the **MDL** is probably better than nothing at all. The MDL can also be used in conjunction with inspection of the **complex roots**.

The SVD Alternative

You will generally find the **Data SVD** or **Nrml SVD** options a better way to approach order selection. When a sufficiently large order is used to effect a good signal noise separation within the eigenmodes, it is often straightforward to graphically select the signal space. Further, the SVD procedures automatically discard the noise eigenmodes, removing the influence of noise in the prediction.

List



The **List Data** option lists a table of the selected criteria. The listing uses the **TableCurve 2D text viewer** facility.

Copy



The **Copy Data to Clipboard** option copies the criteria information to the clipboard. Formats include full precision binary (for spreadsheets such as Excel) and ASCII (for pasting into text editors).

Save



The **Save Data to Disk** option writes the criteria information to a supported file format. These formats include ASCII, Excel 97/2000, Excel 95, Lotus WK3, Lotus WK1, SPSS, or Systat.

Complex Roots

The [Plot Roots](#) option is available in the [AR Modeling and Prediction](#) procedure for inspecting the complex roots of the parametric model.

The inspection of the roots can offer an indication of signal space, similar to that of the [Graphically Select Signal and Noise Sub-Spaces](#) signal selection in the SVD procedures. This inspection can also confirm the stability of the filter arising from the fit.

Pole and magnitude plots are offered. The AR roots are plotted with the + symbol. In an AR fit, the poles on or near the unit circle are usually comprised of signal, while those within the interior of the unit circle are generally noise. In a [Unit Circle](#) plot, all of the roots are plotted. For real data, there will be symmetry across the imaginary axis.

[Magnitude](#) plots are also available. Here the signal components should lie at or near 1.0 while the noise components will have lower magnitudes.

The [AR Out](#) information field reports the number of AR roots that lie outside the unit circle. These roots have a magnitude greater than 1.0. An AR model can be stabilized in the [AR Modeling and Prediction](#) option. Spectral factorization is available to map these roots either to the unit circle or reflectively into the interior of the unit circle.

In general, the [least-squares AR algorithms](#) yield stable filters, but are not assured to do so. Typically, precision losses due to roundoff error generate the instability. Roots with magnitudes that are only slightly above 1.0, as in the fourth decimal place, may be indicative of borderline precision issues attributable to overspecifying the model order or accumulated algorithm roundoff errors. The Data AR algorithms, which directly fit a data matrix, will generally fare better than the Nrml AR algorithms which give up precision when constructing the normal equations matrix.

List



The [List Data](#) option lists the roots in a table containing type, real, imaginary, magnitude, and frequency columns. Only positive frequencies are listed. If the [Unit Circle](#) option is selected, the roots will be sorted by increasing frequency. If the [Magnitude](#) option is selected, the roots will be sorted by decreasing magnitude. The listing uses the [TableCurve 2D text viewer](#) facility.

Copy



The [Copy Data to Clipboard](#) option copies the information in the [List Data](#) option to the clipboard. Formats include full precision binary (for spreadsheets such as Excel) and ASCII (for pasting into text editors).

Save



The [Save Data to Disk](#) option writes the information in the [List Data](#) option to a supported file format. These formats include ASCII, Excel 97/2000, Excel 95, Lotus WK3, Lotus WK1, SPSS, or Systat.

Process Menu



The **Process Menu** is used to start the automated curve-fitting, to customize the equation set for a custom fit, to set the fit preferences, or to install and inspect up to fifteen **user-defined functions** which can be added to the equation set.

Process Menu

-  Curve-Fit All Equations
-  Curve-Fit Linear Equations
-  Curve-Fit Simple Equations
-  Curve-Fit Robust Straight Line
-  Curve-Fit Poly/Ratl Equations
-  Curve-Fit Peak Functions
-  Curve-Fit Transition Functions
-  Curve-Fit Kinetic Equations
-  Curve-Fit Waveform Functions

-  Curve-Fit Custom Equation Set
-  Edit Custom Equation Set
- Read Custom Equation Set

-  Curve-Fit User Functions
-  User Functions
- Last Session User Functions

-  Curve Fit Preferences
-  Background Thread Processing
- TCEX.DLL Functions

Related Topics

Robust Fitting
Chebyshev Series Equations
Fourier Series Equations
High Precision Polynomials and Rationals
Constrained Non-Linear Rationals
Ligand UDF Library

Automated Fitting

You can select from seven predefined equation sets to fit:

- All Equations
- Linear Equations
- Simple Equations
- Robust Straight Line Equations
- Polynomial/Rational Equations
- Waveform Functions
- User Functions Only.

You can also fit three non-linear equation groups, each of which has its own separate configuration. These consist of

- Peak Functions
- Transition Functions
- Kinetic Equations

The **Curve-Fit Custom Equation Set** is used to fit your own custom equation set. Here, the equation groups or individual non-linear equations are selected in the **Edit Custom Equation Set** option.

Linear and non-linear fitting controls for all fitting options, except the Peak, Transition, and Kinetics sets which contain their own separate configurations, are set in the **Curve-Fit Preferences** option.

All active **User Functions** and all active **TCEX.DLL Functions** are always fit, regardless of the equation set being fitted.

Fit Progress

The first two bars reflect the progress of the linear fitting, the first displaying the progress in the computation of the various sums, and the second in fitting the various linear equations. Separate progress bars are shown for the high precision polynomial and rational fitting and for the iteratively fitted rational equations.

The last two bars reflect the progress of non-linear fitting, the first displaying the progress in the sequence of equations to be fitted and the second offers a graphical representation of the r^2 achieved. The r^2 bar is divided into five equal log sections.

If the **Iteration Update** box is checked, the r^2 bar and iteration and r^2 fields are updated with each iteration. Otherwise, the update occurs only when a fit is concluded. Note that this graphical update at each iteration can significantly slow the non-linear fitting. It is recommended that iteration update only be used when fitting is excruciatingly slow, and you want to halt a number of non-promising non-linear fits prior to convergence.

Aborting Automated Fit

If the **Cancel** button is pressed during any portion of the automated fitting, the curve-fit is completely aborted.

Halting a Specific Non-Linear Fit

If **Stop Current Fit** is pressed during the non-linear portion of the fitting, the equation currently being fit is stopped at the current iteration and added to the list of fitted equations in this non-converged condition.

This halt will be immediately signaled and processed.

Automatic Start of Review

If you do not respond to the message reporting the conclusion of the fit, the **Review** will be automatically started with the start option last used to begin the Review.

User-Initiated Background Fitting

Use the **Fit in Background** button to fit the TableCurve equations in the background while working on another Windows task. You can also accomplish this by minimizing the main TableCurve window. You can return to TableCurve at anytime to inspect the progress of the fitting. When the fit is complete, an audible beep will signal the conclusion of the fit. TableCurve performs the fitting on a separate thread. This means you should not have any difficulty working with any other application while the TableCurve fitting is underway.

Background Thread Processing

Although both produce the same numerical results, the automated fitting initiated from the Process menu Curve-Fit options is completely distinct from the **Background Thread Processing** fitting which takes place without user intervention. If a background thread fitting is underway and you select a Process menu Curve-Fit option, the background fit is discarded even if you select the same fit as that underway in the background.

Curve-Fit All Equations

This option begins TableCurve's [Automated Curve-Fit Processing](#) and fits all of TableCurve 2D's equations, all installed [UDFs](#), and all active [TCEX.DLL](#) equations. This option fits 3491 linear equations, 174 non-linear equations, up to 15 installed UDFs, and up to 100 active TCEX.DLL equations.

Fit Controls

 Linear and non-linear fitting controls for this All Equations fitting option are set in the [Curve Fit Preferences](#) option. This is true for the linear and non-linear fitting controls for all fitting options, except the Peak, Transition, and Kinetics sets which contain their own separate configurations.

Note that the intercept setting in the [Curve-Fit Preferences](#) is ignored when fitting All Equations. Both intercept and zero-intercept versions of the non-linear equations are fitted.

Curve-Fit Linear Equations

This option begins TableCurve's [Automated Curve-Fit Processing](#) and fits all of TableCurve's 3491 linear equations, all installed [UDFs](#), and all active [TCEX.DLL](#) equations.

Fit Controls

 Linear and non-linear fitting controls for this Linear Equations fitting option are set in the [Curve-Fit Preferences](#) option. This is true for the linear and non-linear fitting controls for all fitting options, except the Peak, Transition, and Kinetics sets which contain their own separate configurations.

Curve-Fit Simple Equations

This option begins TableCurve's **Automated Curve-Fit Processing** and fits TableCurve's 2-parameter equations, the intercept and non-intercept forms of the non-linear power and exponential functions, all installed **UDFs**, and all active **TCEX.DLL** equations.

Fit Controls



Linear and non-linear fitting controls for this Simple Equations fitting option are set in the **Curve Fit Preferences** option. This is true for the linear and non-linear fitting controls for all fitting options, except the Peak, Transition, and Kinetics sets which contain their own separate configurations.

Curve-Fit Robust Straight Line

This option begins TableCurve's [Automated Curve-Fit Processing](#) and fits a simple straight line equation by all four of the [Minimization Criteria](#) available within TableCurve's non-linear fitting engine. This option also fits all installed [UDFs](#), and all active [TCEx.DLL](#) equations. Note that the standard least-squares minimization is also fit using the non-linear engine to assure an equal precision of convergence for all minimizations. This option is ideal when fitting data to a straight line where outliers are likely or suspected and a robust model is desired to lessen the impact of such outliers upon the overall fit.

Fit Controls



The non-linear fitting controls for this Robust Straight Line fitting option are set in the [Curve Fit Preferences](#) option. This is true for the linear and non-linear fitting controls for all fitting options, except the Peak, Transition, and Kinetics sets which contain their own separate configurations.

Note that the minimization setting in the [Curve-Fit Preferences](#) is not used in this option since all of the robust minimizations are made.

Curve-Fit Poly/Ratl Equations

This option begins TableCurve's **Automated Curve-Fit Processing** and fits all of TableCurve 2D's polynomial and rational functions, all installed **UDFs**, and all active **TCEX.DLL** equations. This option includes the high precision, Chebyshev, and Fourier series polynomials and rationals as well as the non-linearly fit constrained rational models.

Fit Controls



Linear and non-linear fitting controls for this Polynomial/Rational Equations fitting option are set in the **Curve Fit Preferences** option. This is true for the linear and non-linear fitting controls for all fitting options, except the Peak, Transition, and Kinetics sets which contain their own separate configurations.

Specialty Polynomials and Rationals

Chebyshev Series Equations

Fourier Series Equations

High Precision Polynomials and Rationals

Constrained Non-Linear Rationals

Curve-Fit Peak Functions

TableCurve 2D's built-in **peak functions** consist of a set of 74 total equations. There are 37 peak models with and without an intercept. To make peak function fitting as flexible as possible, you are first presented with a peak configuration dialog which contains custom fit settings specifically for the built-in peak functions. You can choose only the **symmetric** or **asymmetric peaks**, one or both of the intercept and zero-intercept versions as well as any one of the four **Minimization Criteria** available in the non-linear fitting engine. You can choose individual **peak functions** or you may have the **AI Expert** feature pick the settings for you based upon an analysis of the data profile. The non-linear fitting controls override the non-linear fit preferences in the **Curve-Fit Preferences** option. Your peak function customization will be automatically saved across sessions.

Also fitted with this option are all installed **UDFs**, and all active **TCEX.DLL** equations.

Symmetric Peak Functions

Basic three-parameter (single width) **symmetric peaks** include the **Gaussian**, **Lorentzian**, **Logistic**, **Complementary Error Function**, and **Laplace** models. Peaks with a second width term that control the decay or shape of the peak include the **Modified Gaussian**, **SDS**, **SDC**, **Pearson VII**, **Gaussian-Lorentzian Cross Product**, **Error**, and **Student t** peak functions. The **Pearson VII**, **Student t**, and **Gaussian-Lorentzian XP** each produce a symmetric peak function that can vary between a Gaussian and Lorentzian. The **Pearson VII** and **Student t** models, which are actually two different parametrizations of the same model, can actually go further, producing a distribution with wider tails than the Lorentzian. In the **Modified Gaussian** and **Error** models, the Gaussian's power of 2 is replaced by an adjustable parameter expression. Both models are capable of producing the Laplace or double-sided exponential. The **SDS** (symmetric double sigmoidal) and the **SDC** (symmetric double cumulative) functions can vary from approximating a Gaussian at one extreme to that of a square wave profile at the other. The **SDS** and **SDC** functions enable fitting profiles where there is a plateau at the peak maximum.

Asymmetric Peak Functions

TableCurve 2D's asymmetric peaks include the statistical distribution functions, the single width **Log-Normal** and **Extreme-Value**, the two-width term **Log-Normal**, **Extreme Value (fronted and tailed)**, **Gamma**, **Weibull**, **Inverted Gamma**, and **Chi-Square**, and the three width term **Beta**, **F-variance**, and **Pearson IV** models. All are parametrized to have an amplitude and peak center consistent with all of TableCurve 2D's peak functions.

The asymmetric functions include two convolution models, the **EMG** (Exponentially Modified Gaussian), and the **GMG** (Half-Gaussian Modified Gaussian). Also included are the **ADS** and **ADC**, asymmetric versions of the SDS and SDC peaks. There are several forms of **asymmetric logistic peaks** as well as the peak formed by the **derivative of the LDR** (logistic dose response) equation. There are three forms of a **pulse peak** for fitting profiles with a very rapid rise and much slower decay. There are also two kinetics peaks, one for a first order **intermediate** and another for a first order **equilibrium**. Unlike the versions in the kinetics equation set which assume the reaction begins at time $X=0$, these two include a lag term.

Synthetic Functions

TableCurve 2D's built-in peak functions have all been constructed to be defined for all values of X , even if the native function has undefined regions or is undefined at certain parameter values. These synthetic functions will either return zero (for zero-intercept versions) or return to the intercept value (for intercept versions) when an undefined condition is present. This is especially useful with peaks such as the **Weibull**, **Gamma**, **Inverted Gamma**, **Log Normal-4**, **Chi-Square**, **Beta**, **F-variance**, and **Pulse** peaks which are undefined below and/or above certain X -values. If you are fitting data that are at zero or a constant value until a peak profile appears starting at a certain X , you need not remove these initial points. They

can be fitted as a part of the actual model.

AI Expert



When selecting this option, TableCurve 2D references raw data measurements of baseline and asymmetry to determine which peaks are most appropriate to the data as well as whether zero-intercept, intercept, or both versions should be fitted. The **AI Expert** option will only be effective if there is a full peak present so that an accurate baseline can be estimated, and only if there are sufficient points near the left and right half-maxima points, since the asymmetry determination is made at this half-maxima value.

Reading and Saving Peak Fitting Configurations

The most recent peak configuration is saved across sessions and presented each time this option is used. If you select the AI Expert and are not satisfied with the suggested fit, you can press **Reset** to restore the configuration present when first opening this dialog. If you will be fitting several types of specific peak profiles, you may want to save a configuration specific to each or to a given application or problem. The **Save** item saves all information shown in the dialog to a binary file with an [FPK] extension. The **Read** item can then be used to access this configuration whenever desired.

Nomenclature

During fitting, a function name with an underscore suffix, such as Gaussian_, represents the zero-intercept version of the function. When the underscore is absent, the intercept version of the function is being referenced. In the Review, non-linear equations can be presented in either a full mathematical representation or in a symbolic notational format. Since some of the peak functions have rather long mathematical expressions, you may prefer to use the notational format offered in the **Preferences** option in the main Edit menu or in the File menu of the Curve-Fit graph.

Automated Fitting



Upon closing the peak configuration via the **Fit** button, TableCurve 2D's **Automated Curve-Fit Processing** begins and fits the peak functions as configured as well as any installed UDFs and TCEX.DLL functions. The non-linear fitting controls in the peak configuration also apply to such UDFs and external functions.

Curve-Fit Transition Functions

TableCurve 2D's built-in **transition functions** consist of a set of 29 total equations. There are 14 **transition models** with and without an intercept and a special impulse transition model. To make transition function fitting as flexible as possible, you are first presented with a transition configuration dialog which contains custom fit settings specifically for the built-in **transition functions**. You can choose only the **symmetric** or **asymmetric transitions**, one or both of the intercept and zero-intercept versions, as well as any one of the four **Minimization Criteria** available in the non-linear fitting engine. You can choose individual **transition functions** or you can have the AI Expert feature pick the settings for you based upon an analysis of the data profile. The non-linear fitting controls override the non-linear fit preferences in the **Curve-Fit Preferences** option. Your transition function customization will be automatically saved across sessions.

Also fitted with this option are all installed **UDFs**, and all active **TCEX.DLL** equations.

Symmetric Transition Functions

TableCurve 2D's symmetric transition functions include the **Sigmoid**, the **Gaussian cumulative**, the **Lorentzian cumulative** and the **SDS (symmetric double sigmoidal) cumulative**. The **Lorentzian cumulative** is the least sharp of TableCurve 2D's basic symmetric transitions and the **Gaussian cumulative** is the sharpest. The **SDS cumulative** has an additional width term enabling it to model most symmetric transitions, including those with a long straight line section.

Asymmetric Transition Functions

TableCurve 2D's asymmetric transition functions include the **Logistic Dose Response** model, the **Log-Normal cumulative**, the **Extreme-Value cumulative**, and the **Weibull cumulative**. Also available is the **Pulse cumulative** and the **Pulse cumulative with a variable power term**. Both fit transitions with a very rapid rise and much slower decay. The **Cascade transition** offers a similar profile, this based on formation in sequential first order kinetics. Unlike the sequential formation in the kinetics equation set which assumes the reaction begins at time $X=0$, the Cascade transition function includes a lag term. Also offered are two **asymmetric sigmoids** and an **impulse transition** which can model near instantaneous transitions within data whose Y-values are continuously increasing or decreasing with increasing X. The function set also includes the **cumulative for the EMG (Exponentially-modified Gaussian)**.

Synthetic Functions

TableCurve 2D's built-in transition functions have all been constructed to be defined for all values of X, even if the native function has undefined regions or is undefined at certain parameter values. These synthetic functions will either return zero (for zero-intercept versions) or the intercept value (for intercept versions) when an undefined condition is present. This is especially useful with the Weibull and Pulse transitions. If you are fitting data that are at zero or a constant value until the transition profile begins, you need not remove any initial points. They can be fitted as a part of the actual model.

Negative Widths

Some transition functions produce a true reverse transition when the sign is changed on the width term. For these functions, TableCurve will use this mathematical symmetry to enable the fitting of a zero-intercept descending transition. For those transition functions which do not exhibit this property, descending transitions can only be fit with intercept forms.

AI Expert



When selecting this option, TableCurve 2D references raw data measurements of upper and lower

baseline and asymmetry to determine which transition functions are most appropriate to the data as well as whether zero-intercept, intercept, or both versions should be fitted. The AI Expert option will only be effective if there is a full transition present so that accurate baselines can be estimated and only if there are sufficient points within the transition, since the asymmetry determination is made at the transition center.

Reading and Saving Transition Fitting Configurations

The most recent transition configuration is saved across sessions and presented each time this option is used. If you select the AI Expert and are not satisfied with the suggested fit, you can press [Reset](#) to restore the configuration present when first opening this dialog. If you will be fitting several types of specific transitions, you may want to save a configuration specific to each or to a given application or problem. The [Save](#) item will save all information shown in the dialog to a binary file with an [TRN] extension. The [Read](#) item can then be used to access this configuration whenever desired.

Nomenclature

During fitting, a function name with an underscore suffix, such as Sigmoid_, represents the zero-intercept version of the function. When the underscore is absent, the intercept version of the function is being referenced. In the Review, non-linear equations can be presented in either a full mathematical representation or in a symbolic notational format. Since some of the transition functions have rather long mathematical expressions, you may wish to use the notational format offered in the [Preferences](#) option in the main Edit menu or the File menu of the Curve-Fit graph.

Automated Fitting



Upon closing the transition configuration via the [Fit](#) button, TableCurve 2D's [Automated Curve-Fit Processing](#) begins and fits the transition functions as configured as well as any installed UDFs and TCEX.DLL functions. The non-linear fitting controls in the transition configuration will also apply to such UDFs and external functions.

Curve-Fit Kinetic Equations

TableCurve 2D's built-in **kinetics equations** consist of a set of 58 total equations where the reaction begins at time $X=0$. To make kinetics fitting as flexible as possible, you are first presented with a kinetics configuration dialog which contains custom fit settings specifically for the built-in **kinetic equations**. You can choose **decay** or **formation equations**, one or both of the intercept and zero-intercept versions as well as any one of the four **Minimization Criteria** available in the non-linear fitting engine. You can choose individual **kinetics equations** or for decay or formation profiles, you can have the AI Expert feature pick the settings for you based upon an analysis of the data. The non-linear fitting controls override the non-linear fit preferences in the **Curve-Fit Preferences** option. Your kinetic equations customization will be automatically saved across sessions.

Also fitted with this option are all installed **UDFs**, and all active **TCEX.DLL** equations.

Simple Decay and Formation $A \rightarrow B$

The decay (Concentration of A) or formation (Concentration of B) is covered for orders $1/2$, 1 , 2 , 3 , and n . In the **order n equations**, the order is an adjustable parameter. For an effective determination of the order n , it is recommended that the data have a zero baseline and that the zero intercept version of this function be fitted. For each order there is both an intercept and zero-intercept version as well as both a formation and decay version. There are thus 20 basic built-in kinetics equations, all of which compute true rate constants.

Simple Decay and Formation $A \rightarrow B$ Hyperbolic Forms

The second order decay or formation is often given using a **hyperbolic form** rather than a true second order form. TableCurve 2D also includes intercept and zero-intercept forms of both decay and saturating hyperbolas. These four equations produce identical goodness-of-fit statistics as the natural second order equations, although the rate constant will not be the actual k in the differential equation defining the reaction.

Decay and Formation $A \rightarrow B$ Simultaneous 1st, 2nd Order

This equation models reactions whose mechanisms have a **simultaneous first and second order** component. Here there is a single term associated with maximum concentration and two rate constants, one for the 1st order reaction pathway and the other for the second order pathway. TableCurve 2D offers four equations, intercept and zero-intercept forms of both the decay and formation profiles.

Formation $A \rightarrow B \rightarrow C$, 1st, 1st Order

This equation models **sequential formation** (the concentration of C) when $A \rightarrow B$ is via a first order mechanism and $B \rightarrow C$ is also first order. Here there are a single maximum concentration term and the two separate first order rate constants. Note that this kinetic model is identical to the Cascade transition function in the transition equation group except that it assumes the reaction begins at time $X=0$ (the Cascade transition contains a lag term). There are both intercept and zero-intercept forms.

Decay $A+B \rightarrow C$, 1st Order

These four equations model the **decay of A via a first order mechanism**. There are both intercept and zero intercept forms for the case when the molar concentration of A is greater than B as well as when the molar concentration of B is greater than A.

Decay $A \rightarrow B$, $A \rightarrow C$ and Formation $A \rightarrow C$, $B \rightarrow C$

These twelve equations represent the sum of two separate decays or two separate formations. Four equations are based on **both being first order**, four on **both being second order**, and four upon **one reaction being first order and the other second order**. These will probably be the models of choice for data that are well fitted by the sum of two exponentials or the sum of two hyperbolas.

Equilibrium $A=B$, 1st order forward and reverse

TableCurve 2D offers two forms for the standard first order equilibrium concentration of A. One form computes the **initial concentration and the forward and reverse rate constants**, while the other determines the **initial and equilibrium concentrations and the net sum of the rate constants**. Both forms will produce identical goodness-of-fit statistics.

Complex Equilibrium $A=B+C$ and $A+B=C+D$

TableCurve 2D offers models for fitting the complex equilibrium reactions $A=B+C$ and $A+B=C+D$. Both assume first order forward and reverse kinetics and both fit the concentration of A. These equations fit an initial concentration, an equilibrium concentration, and a net rate constant.

Intermediate Equations $A \rightarrow B \rightarrow C$

TableCurve 2D offers **four equations for fitting the concentration of B**. These functions are based upon first order kinetics and produce peak profiles which fully decay. There are intercept and non-intercept forms for the case of the $A \rightarrow B$ rate constant being greater than the $B \rightarrow C$ rate constant, and for the case where the $A \rightarrow B$ rate constant is smaller. Both cases will produce identical goodness-of-fit statistics. Note that this kinetic model is identical to the Intermediate peak function in the peak equation group except that it assumes the reaction begins at time $X=0$ (the Intermediate peak function contains a lag term).

Equilibrium Intermediate Equations $A \rightarrow B \rightleftharpoons C$

TableCurve 2D offers **four equations for fitting the concentration of B**. These functions are based upon first order kinetics and produce peak profiles which decay to an equilibrium concentration. There are three rate constants, two forward and one reverse. There are intercept and non-intercept forms for the case of the $A \rightarrow B$ rate constant being greater than the $B \rightleftharpoons C$ forward rate constant, and for the case where the $A \rightarrow B$ rate constant is smaller. Both cases produce identical goodness-of-fit statistics. Note that this kinetic model is identical to the Equilibrium peak function in the peak equation group except that it assumes the reaction begins at time $X=0$ (the Equilibrium peak function contains a lag term).

Michaelis-Menten Equation

The **Michaelis-Menten model** is commonly used to describe the saturable kinetics which occur in many enzyme studies. This model can be readily fit using the **IMPLICIT** function within a UDF. TableCurve 2D supplies **MICHMENT.UDL**, which can be used to fit this implicit model. The library also contains an explicit UDF solution. The initial estimates will be automatically determined, provided sufficient data exists for TableCurve 2D to accurately map a transition center and width.

Synthetic Functions

TableCurve 2D's built-in kinetics functions have all been constructed to be defined for all values of X , even if the native function has undefined regions or is undefined at certain parameter values. These synthetic functions will either return zero (for zero-intercept versions) or the intercept value (for intercept versions) when an undefined condition is present. Note that this in no way alters the requirement that data begin at time $X=0$.

AI Expert



Select this option only for simple decay and formation profiles. The [AI Expert](#) option will reference raw data measurements to determine whether decay or formation versions are appropriate as well as whether zero-intercept, intercept, or both versions should be fitted.

Reading and Saving Kinetics Fitting Configurations

The most recent kinetic equation configuration is saved across sessions and presented each time this option is used. If you select the AI Expert and are not satisfied with the suggested fit, press [Reset](#) to restore the configuration that was present when first opening this dialog. If you will be fitting several types of kinetic data, you may want to save a configuration specific to each or to a given application or problem. The [Save](#) item will save all information shown in the dialog to a binary file with an [KIN] extension. The [Read](#) item can then be used to access this configuration whenever desired.

Nomenclature

During fitting, a function name with an underscore suffix, such as Decay2_, represents the zero-intercept version of the function. When the underscore is absent, the intercept version of the function is being referenced. The function names discriminate between decay and formation forms. In the Review, non-linear equations can be presented in either a full mathematical representation or in a symbolic notational format. You may want to use the notational format offered in the [Preferences](#) option in the main Edit menu or the File menu of the Curve-Fit graph.

Automated Fitting



Upon closing the kinetics configuration via the [Fit](#) button, TableCurve 2D's [Automated Curve-Fit Processing](#) begins and fits the kinetic equations as configured, as well as any installed [UDFs](#) and [TCEX.DLL](#) functions. The non-linear fitting controls in the kinetics configuration will also apply to such UDFs and external functions.

Curve-Fit Waveform Functions

This option begins TableCurve's **Automated Curve-Fit Processing** and fits TableCurve 2D's **waveform functions** by the full non-linear procedure. These include both intercept and zero-intercept versions of the **sine and sine-squared equations**. Also fitted are all installed **UDFs**, and all active **TCEX.DLL** equations.

Fit Controls

 Non-linear fitting controls for this Waveform Equations fitting option are set in the **Curve Fit Preferences** option. This is true for the linear and non-linear fitting controls for all fitting options, except the Peak, Transition, and Kinetics sets which contain their own separate configurations.

Curve-Fit Custom Equation Set

This option begins TableCurve's [Automated Curve-Fit Processing](#) and fits a custom equation set. This set will either be the default custom set, the set from a prior session if this default set was ever modified, a new set installed via the [Read Custom Equation Set](#) Process menu item or a new set configured via the [Edit Custom Equation Set](#) Process menu item. Use the [Edit Custom Equation Set](#) item to view or change the current custom fit settings.

Fit Controls

 Linear and non-linear fitting controls for this custom fitting option are set in the [Curve-Fit Preferences](#) option. This is true for the linear and non-linear fitting controls for all fitting options, except the Peak, Transition, and Kinetics sets which contain their own separate configurations.

Edit Custom Equation Set

This option is used to configure a custom equation set for TableCurve 2D's [Automated Fitting](#). You can select linear families of equations or specific non-linear equations to be included in the equation set that is fitted. The equation selections apply only to the [Curve-Fit Custom Equation Set](#) fitting option.

Equation Selection

When you check a linear selection in the [Standard](#), [Polynomial](#), and [Rational](#) groupings, you are selecting an equation family. Some families have only a small number of equations while others contain a very large number. The TableCurve 2D equation families are listed in the appendix of the manual and can be viewed by using the F2 [Equation Help](#) item in the main menu's Help.

When you check a [non-linear equation](#) in the [Peak](#), [Transition](#), [Kinetic](#), or [Other](#) groupings, you are selecting a specific model. In many cases, a non-linear model consists of both intercept and non-intercept forms. The intercept type is set in the [Curve-Fit Preferences](#), available directly from the dialog via the [Options](#) button. Note that the number of non-linear equations will depend on whether only one or both of the Intercept or Zero-Intercept versions are selected.

The TableCurve 2D non-linear equations are listed in the non-linear equations section of the manual and can be viewed by using the F2 [Equation Help](#) item in the main menu's Help.



Note that if you are only fitting [Peak Functions](#), only [Transition Functions](#), or only [Kinetic Equations](#), separate custom configurations are available in the [Curve-Fit Peak Functions](#), the [Curve-Fit Transition Functions](#), and the [Curve-Fit Kinetic Equations](#) Process menu items.

Many of the groups contain an [All](#) button that selects all of the models or families and a [Clear](#) button that unchecks all models or families currently selected.

Fit Controls

Linear and non-linear fitting controls for fitting this custom equation set are configured in the Process menu's [Curve-Fit Preferences](#) option. To set these controls from this custom equation set dialog, click the [Options](#) button.

Fitting this Custom Set

Use the [Fit](#) item to immediately fit the equation set within this dialog. This is the same as using the [Curve-Fit Custom Equation Set](#) Process menu option.

Saving a Custom Equation Set

Use the [Save](#) item to save the current equation set to disk. The default file extension is FIT. These are binary files that can only be produced within the program. Custom fit files can be recalled at anytime. The current custom set is always saved across sessions. Save custom sets to disk if you plan to use more than one custom set in your work.

Reading a Custom Equation Set

Use the [Read](#) item to read a custom equation set. The current settings will be updated to reflect this new set. You may also use the [Read Custom Equation Set](#) Process menu option to read a custom set.

Reset

The [Reset](#) button simply restores the settings to the state that existed at the time the dialog was opened.

Read Custom Equation Set

This item reads a custom equation set from disk, enabling this new configuration to be immediately fitted using the [Curve-Fit Custom Equation Set](#) item in the Process menu. To view the settings installed by reading the custom fit file, use the [Edit Custom Equation Set](#) Process menu item.

Curve-Fit User Functions

This option begins TableCurve 2D's **Automated Curve-Fit Processing** and fits TableCurve's **UDFs**, if installed. Also fitted are all active **TCEX.DLL** equations.

Fit Controls

 Non-linear fitting controls for this UDF fitting option are set in the **Curve Fit Preferences** option. This is true for the linear and non-linear fitting controls for all fitting options, except the Peak, Transition, and Kinetics sets which contain their own separate configurations.

User-Defined Functions

You may install up to fifteen non-linear User-Defined Functions (UDFs) into TableCurve's equation set at any given time. This [User-Defined Functions](#) option is used as the control center for entering, editing, saving, and fitting UDFs. From this UDF dialog, you can work with individual functions or with UDF libraries. A TableCurve UDF contains all information necessary to fit the function, including the function name, the parameter count, the function's formula, and starting estimates and constraints for each parameter. A special Adjust item allows you to graphically adjust the starting estimates to better assure a successfully converged fit. You may also inspect the partial derivatives for the UDF to find instances of multiple constants, insignificant parameters, and to expose conditions where a fit would be likely to fail.

UDF Selection

The keypad with buttons labeled 1 through 15 is used to select the current UDF. In TableCurve's non-linear equation set, UDF#1 is Equation 8001, UDF#2 is Equation 8009, and UDF#3 through UDF#15 are Equations 8017-8029. UDFs active at any given time need not be sequential and they will remain in the position where you install them, even if lower number UDFs are empty.

Function Name

This name will be used to represent the function in the curve-fit graphs. You will wish to select a name that is meaningful to you.

Coefficient Count

This is the number of adjustable parameters in the UDF model. The UDF must contain the number of parameters entered. If, for example, you enter 4 as the coefficient count, the UDF must contain parameters #A,#B,#C,#D (or A0,A1,A2,A3). The maximum coefficient count is 10.

Entering the UDF

The UDF entry uses a simple ASCII multiline editor. You can use the Cut, Copy, and Paste items to move text about or to paste in the UDF formula if you placed it into the clipboard via another program. All of the functions and constants available within TableCurve can be accessed via a special [Function Insert](#) help. In a UDF no prefixes are automatically supplied.

You may define as many constants as you wish. For example, $\text{SQRT2PI}=\text{SQRT}(2*\text{PI})$ would be defined on one line and used in subsequent lines. Constant expressions are evaluated once and the numeric result is stored. Any assignment to a variable other than F1-F24 (or #F1-#F24) and Y is assumed to be a constant.

Any expression containing X or any of the adjustable parameters #A-#J (or A0-A9) must be assigned either to an F1-F24 expression or to Y. F1-F24 and Y expressions are compiled and are evaluated once for each data point in every iteration. The Y expression must always be the last line in the UDF. Here is a simple UDF example, first with the #A-#J nomenclature and then with the A0-A9 format:

```
S2=SQRT(2)
F1=ERFC(-#D/S2+LN(X/#C))/(#D*S2)
Y=#A+0.5*#B*F1
```

```
S2=SQRT(2)
F1=ERFC(-A3/S2+LN(X/A2))/(A3*S2)
Y=A0+0.5*A1*F1
```

The S2 expression is a constant that is evaluated numerically and stored for use in subsequent

expressions. The F1 expression calls the complementary error function with an argument expression containing the third (#C or A2) and fourth (#D or A3) adjustable parameters. The last line is the required Y= function expression. It contains the first (#A or A0) and second (#B or A1) adjustable parameters. The F1 and Y expressions are compiled. In the fit, these expressions are evaluated at the X of every data point for each iteration's particular adjustable parameters.

UDFs Based on a TableCurve Built-in Equation

It is not necessary to manually enter any of TableCurve's built-in equations in a UDF. TableCurve can **Generate UDFs** for built-in equations with 10 or fewer parameters. The **Save TableCurve UDF** option found in the File menu of the Review Curve-Fit graph is used to generate such UDFs. A generated UDF will consist of the full mathematical equation expression rather than a reference to a built-in function. The estimates in the generated UDFs will consist of the values of the actual fitted parameters. The following generated UDF expression is an example for the Weibull function.

```
F1=(X-#B)/#C
F2=(#D-1.0)/#D
Y=#A*EXP(-(F1+F2^(1.0/#D))^#D+F2)*F2^(-F2)*(F1+F2^(1.0/#D))^(#D-1.0)
```

Directly Accessing TableCurve's Non-Linear Functions

For non-linear equations, generated UDFs will not execute as fast as the built-in functions available via the **Non-Lin** function insert help in the UDF entry dialog. The built-in non-linear functions also contain the conditional code insuring that the functions will return a valid value for all X whereas the generated UDFs will contain only the native equation without conditional statements. Note that the built-in non-linear functions available in UDFs are zero-intercept versions when both a zero intercept and intercept form exists. In UDFs where you use a built-in non-linear function, you will have to add an intercept term if such is desired. The following UDF expression is the equivalent of the above generated Weibull function.

```
Y=WEIBULL_(A0,A1,A2,A3)
```

Indirect Function References

The following functions utilize an indirect reference for the initial argument:

```
DX(n)
DX2(n)
AI(n,st,end)
AIP(n,st,end,prec)
QIP(n,st,end,prec)
SUM(n,st,end,inc)
SER(n,st,inc,lim)
PROD(n,st,end,inc)
IMPLICIT(n,yl,yh,prec)
```

In versions prior to v4.02, n had to be specified as an integer from 1 to 9. TableCurve 2D now supports entering Fn and #Fn in the initial argument position. For example, if F3 were to be processed by any one of these functions, the first argument could be 3, F3, or #F3.

UDFs with Derivative Functions

The first derivative function is **DX(n)** where n is from 1 to 9 and references an F1 to F9 expression. The second derivative function is **DX2(n)**. The following example is for the first derivative of a Gaussian:

```
F1=A0*EXP(-0.5*((X-A1)/A2)^2)
```

Y=DX(1)

UDFs with Integration Functions

The primary integration function is **AIP**(n,start,end,prec). It first seeks to achieve the target precision using a successive step Gaussian Quadrature. If this is unsuccessful, a Romberg procedure follows. The following example uses the AIP() function to fit the cumulative of the Log-Normal distribution:

```
UPPER=40.0
F1=LN($/#C)/#D
F2=EXP(-0.5*F1*F1)
Y=#A+#B*AIP(2,X,UPPER,1E-6)
```

Note that the \$ symbol is used as the variable of integration. In this example, the second function expression is integrated with \$ ranging from X to this constant upper limit of 40 to a precision of 6 significant figures.

If you wish to limit the integration only to the much faster Gaussian Quadrature procedure, the **QIP** function can be used in place of **AIP**. Both functions return the integration for the best achieved precision when the target precision is not achieved. For the Gaussian Quadrature this will be a 128 step procedure. For the Romberg, this will be either 131072 or 177147 steps, depending on the type of integral.

UDFs with Implicit Functions

The **IMPLICIT** function is used to create models that must be implicitly solved for Y. The following UDF example is for the Michaelis-Menten model (**MICHMENT.UDL**):

```
F1=IF((Y.EQ.DATAERR).OR.(Y.LE.0),0,A0+A2*LN(A0)-A1*X-A2*LN(Y)-Y)
Y=IMPLICIT(F1,YMIN,YMAX,1E-8)
```

The following example is for the basic ligand binding model (**LIGAND.UDL**):

```
F1=10^X
F2=((A0*A1/(1+A0*(F1-Y*F1))+A2)*(F1-Y*F1))/F1-Y
Y=IMPLICIT(F2,YMIN,YMAX,1E-8)
```

Note that the Y= is not a part of the expression to be solved. The Fn is implicitly solved for Y assuming the Fn evaluates to 0. If your implicit function is of the form $y=f(x,y)$ or $1=f(x,y)$, you must be sure to subtract Y or 1 from the Fn expression.

Validation

A UDF is extensively validated before it is compiled. If there is a math or parser error, you will be given a clear indication of the error and the cursor will be placed at the location where the validation failed. If you are having trouble with a UDF, you may save the failed UDF to disk. You can then recall it at some future time in an effort to fix what is wrong.

Starting Estimates and Constraints

You must enter starting estimates for the parameters in the model. Good starting estimates are sometimes necessary for convergence, and usually result in faster non-linear fits. Minimum and maximum constraints are optional. If a parameter violates a constraint that you set, a penalty function is added to the chi-square to attempt to bring the parameter back into a valid range.

Entering Formulas for Estimates and Constraints

You may enter formulas for the estimates and lower and upper limits in UDFs. These enable a UDF to be

constructed so as to accommodate widely varying X and Y data ranges. Since these formulas are evaluated sequentially, any given formula can reference a previous estimate. For example, #D can reference #A, #B, or #C, but not #E and higher. More commonly, a UDF estimate formula will reference an X-Y data table constant which TableCurve computes for each data set. These can consist of the basic X-Y data table constants, such as XMEAN, XSTD, YRANGE, etc. as well additional constants used specifically for UDFs.

For peak functions, these constants include:

[XCTR](#), X at Peak Center
[XL50](#), X at Half-Maxima Left
[XR50](#), X at Half-Maxima Right
[XW50](#), X Width at Half-Maxima (FWHM)

For transition functions, the following constants are available:

[X50](#), X at $Y_{min} + Y_{range}/2$
[X25](#), X at $Y_{min} + Y_{range}/4$
[X75](#), X at $Y_{min} + 3*Y_{range}/4$
[XWTR](#), X transition width $X_{75} - X_{25}$

For waveform functions, there are:

[XWL](#), wavelength
[XPH](#), phase for sine
[XPH2](#), phase for sine-squared wave

When a UDF contains one or more formulas for the adjustable parameter estimates and limits, it is saved in a binary rather than ASCII form. These UDFs can only be edited inside of TableCurve 2D. The [.UDF] extension is used for both the binary and ASCII formats.

Reading a UDF

Use the [Read](#) button to read a UDF from disk. The UDF will be read into the current UDF position, replacing any UDF that might currently be in this same position. You may read any UDF into any of the 15 available positions, even if lower numbered positions are empty. This option will read both the ASCII format UDFs and the binary format UDFs containing one or more formulas for estimates or constraints. It is recommended that UDFs be created only within the program.

Clearing a UDF

Use the [Clear Current UDF](#) button to immediately free the currently selected UDF from memory and to reset the UDF entry screen.

Saving a UDF

Use the [Save](#) button to save a UDF to disk. If the estimates and constraints are numeric, the UDF is saved in an ASCII format with a [.UDF] extension. If formulas are present, the UDF is saved in a binary format that requires approximately 4K of disk space. A UDF is always validated before a Save is made. If the validation fails, you are given an option to save the UDF even though it contains an error. You can then recall it at some future time in an effort to repair what is wrong.

Saving a UDF Library

To make it easy to work with up to 15 user functions, any set of installed UDFs can be saved as a user function library. UDF positions are preserved as you enter them and installation of empty UDF positions

are permitted. UDF libraries are saved as binary [UDL] files, whether or not estimates contain formulas. In a UDF library, each UDF consumes about 4K, so a full 15 UDF set will require about 60K disk space. To save the current set of UDFs to a library, use the [Save UDF Library](#) button. For maximum flexibility, you may wish to save individual UDFs as separate files in addition to having them in libraries. If you choose to keep UDFs only in libraries, you can save out individual UDF components simply by reading the UDL library, selecting the UDF of interest, and then using the individual [Save](#) option.

Reading a UDF Library

To read the user functions within a TableCurve UDF library, use the [Read UDF Library](#) button. This will install and validate all of the UDFs in the UDL library. If a validation fails, you are notified of such and that specific function is not installed. To extract a UDF from a library, use this option, select the UDF desired, and then use the individual [Save](#) option to create the UDF file.

Clearing All UDFs

To clear all of the currently installed UDFs from memory and to clear the current entry screen, use the [Clear All UDFs](#) button. You must confirm this option before the UDFs are freed. Note that it is not necessary to clear the existing UDFs when reading a UDF library since all current UDFs are cleared prior to this read operation.

Copying UDFs

The [Copy ->](#) button is used to copy the current UDF to the next empty UDF position. This option is particularly helpful when you wish to create a UDF library consisting of similar models.

Immediate Fit

Use the [Fit UDFs](#) button to immediately fit the UDFs. This is the same as the Curve Fit User Functions item in the Process menu. Note that this option fits all installed UDFs. The fitting only occurs if the current UDF is successfully validated.

Parameter Contribution Warning

Even with a successful UDF compilation, you may still get the warning [Parameter n Makes Less Than .1% Fractional Contribution to Equation in X-Range of Data. Adjustment is Recommended](#). TableCurve determines the minimum and maximum partial derivative for each parameter. This range of partial derivative is multiplied by the current estimate for that parameter and then compared to the overall Y-data range. If a parameter does not make at least a .1% contribution relative to the Y-data range, this warning is given.

This warning is often a good indicator that the fit will fail because of poor starting estimates. It may also indicate a poorly designed model that contains a parameter that minimally impacts the equation in the X-range of the data. When this message occurs, it is recommended that you use the [Adjust](#) procedure to refine the estimates, and if necessary, the Adjust procedure's [Derivatives](#) option to see if an insignificant parameter is present.

If You are Uncertain of Starting Estimates

When working with a new or exploratory model, you may not know what values constitute good starting estimates. In such a case, for each parameter you are uncertain of, simply enter 1.0 for the starting estimate (or some other value for which the UDF is defined). If you achieve a successful compile but get the parameter contribution warning, use the [Adjust](#) item to graphically set the estimates.

If you do not get the parameter contribution warning, there is a good chance the fitting algorithm may

successfully converge. There is no harm in letting the fitting algorithm do the work. Use the [Fit UDFs](#) option to fit only the UDFs. If you are successful, you may wish to take note of the successful parameters and modify the UDF accordingly. If the fit fails, you must return to the UDF screen and use the [Adjust](#) feature to graphically set the starting estimates to better reflect the data.

To assist in this adjustment, you may wish to use the [Find and Update](#) option. This performs a limited convergence fit with a subset of the data and automatically places the parameters in the starting estimate fields. In rare cases, this reduced data set fit can wander off into oblivion producing meaningless estimates. If this occurs simply use the [Reset](#) button or Cancel out of the Adjust dialog to discard the modified estimates. In most cases, the Find and Update should produce excellent estimates. If there is a revision of a parameter initially entered as a formula in the main UDF dialog, either by manual adjustment or via the Find and Update option, you must confirm that the original formula for the estimate is to be replaced with the new numeric value. Note that any save operations occurring after this replacement will use this numeric value.

Graphical Adjustment of Starting Estimates

Use the [Adjust](#) item to open a graph of the X-Y data and the UDF. Do not be surprised if you do not see the UDF at first. If the estimates are too far off, it is possible that no part of the UDF graph will be anywhere within the field of the X-Y data.

The goal of graphical adjustment is to set the parameters so that the UDF starts to approximate the data. You may adjust each parameter with the scrollbar or you may enter the actual value. If you enter an actual value, there will be a brief delay before the screen is updated, allowing you to complete your entry.

For each parameter that has user-specified minimum and maximum constraints, as opposed to the defaults, the scrollbar adjustment will range from this minimum to maximum in 100 steps.

If such constraints are absent, the scrollbar's sensitivity will depend on computed partial derivatives. The partial derivatives are updated with each value entered. If you enter a value for which the partial derivative range is very nearly zero, or if such is true when you begin, the scroll bar adjustment may produce very large jumps. In this case you must enter a value that brings the UDF back into range.

Fn Scaling

If the starting estimates are so far off that the UDF curve doesn't appear with the data, you can open the [Scaling](#) option for the graph and change the Y priority to [Fn](#). This should allow you to at see the actual UDF, provided it is defined within the X range of the data. Once you can adjust the parameters to bring the UDF into the Y range of the data, the Find and Update option can often refine the estimates automatically.

Find and Update

The [Find and Update](#) button in the [Adjust](#) screen initiates a low convergence fit using a subset of the current data table. The results of this very rapid pre-fit are automatically placed within the starting estimate fields. While this is often a very attractive means for refining the starting estimates, the algorithm is as vulnerable to local minima as the main non-linear fitting engine. In rare cases, this reduced data set fit can wander off into oblivion producing meaningless estimates. If this occurs simply use the [Reset](#) button or Cancel out of the Adjust dialog to discard the modified estimates.

UDF Failures

The two most common causes of UDF failures are having the UDF produce essentially a constant value across the X-range of the data, or to have a parameter set at such a value, that its impact totally masks the impact of X varying across the range of the data. You have to be especially cautious when using a

UDF with a very narrow range of X-data.

As you adjust the starting estimates graphically, the UDF is being updated. When you return to the UDF screen, the estimates will reflect those set in the graphical procedure. By careful attention in the [Adjust](#) feature, no valid UDF should fail to produce the intended fit.

Inspecting Partial Derivatives

From the graphical UDF Adjustment screen, you can choose the [Derivatives](#) option. This displays a similar set of adjustment controls except instead of seeing a graph, you see a table of partial derivative information.

The % of Y column consists of the % of the Y data range represented by the range of the partial derivative multiplied by the estimate. The last column reports a relative percent based on all non-constant parameters, and these sum to 100%.

The fitting algorithm relies on changes in partial derivatives to guide the revisions in the parameters that occur with each iteration. If a partial derivative is constant, then it should have a value of 1.0, meaning that it is a true constant in the expression. If you see two constants, then at least one of these parameters should be removed. If you find that across wide adjustments of a given parameter there is no significant contribution to the Y-range of the data, whereas the others are making such a contribution, you should seriously question whether this parameter belongs in your model.

Note that you can set the estimate of a given parameter so far out of range that no legitimate partial derivatives are found for any parameter in the X range of the data.

Built-In Functions



TableCurve 2D offers extensive mathematical function support in its [Generate Data](#), [Calculation](#), [View Function\(X\)](#), and [User-Defined Function](#) options.

Function Insert Help

A special [Function Insert](#) help exists for accessing the various functions used within user entered numeric expressions. Click on the type of function you are seeking and then on the specific function of interest. The function is automatically inserted into the calculation, view function, or UDF at the current cursor position. You may need to modify the symbols used as arguments in the function to match the variable intended for the expression.

General Functions

$\wedge$	Power
%	Modulus Division
!	Factorial (of Integer)
ABS(X)	Absolute Value
CEIL(X)	Integer Above X
EXP(X)	Exponential
FLOOR(X)	Integer Below X
FRACTION(X)	Fractional Part of X
INTEGER(X)	Integer Part of X
LN(X)	Natural Logarithm
LOG(X)	Base 10 Logarithm
PI	3.14159265358979323846
RAND(N)	Random Integer Between 0 and N Inclusive
RANDOM	Random Number Between 0 and 1
ROUND(X)	Round X to Nearest Integer
SQRT(X)	Square Root of X

Conditional Expressions

IF(expr,n1,n2)	Evaluates to n1 if expr true, n2 if expr false
> .GT.	Greater
< .LT.	Less
>= .GE.	Greater or Equal
<= .LE.	Less or Equal
== .EQ.	Equal
<> .NE.	Not Equal

Trigonometric Functions

ACOS(X)	ArcCosine of X (radians)
ACOSH(X)	Hyperbolic ArcCosine
ASIN(X)	ArcSine of X (radians)
ASINH(X)	Hyperbolic ArcSine
ATAN(X)	ArcTangent of X (radians)
ATAN2(X,Y)	ArcTangent of X/Y (radians)
ATANH(X)	Hyperbolic ArcTangent

COS(X)	Cosine of X radians
COSH(X)	Hyperbolic Cosine
COT(X)	Cotangent of X radians
COTH(X)	Hyperbolic Cotangent
CSC(X)	Cosecant of X radians
CSCH(X)	Hyperbolic Cosecant
DTOR(Deg)	Degrees to Radians
RTOD(Rad)	Radians to Degrees
SEC(X)	Secant of X radians
SECH(X)	Hyperbolic Secant
SIN(X)	Sine of X radians
SINH(X)	Hyperbolic Sine
TAN(X)	Tangent of X radians
TANH(X)	Hyperbolic Tangent

Statistical Functions

!	Factorial (of Integer)
B0(N)	Bernoulli Number B0(N)
BET(A,B)	Beta Function
ERF(X)	Error Function
ERFC(X)	Error Function, Complement
ERFP1(X)	Error Function + 1.0
ERFM1(X)	Error Function - 1.0
GAM(X)	Gamma Function
HYPM(A,B,X)	Hypergeometric M
HYPU(A,B,X)	Hypergeometric U
HYPF(A,B,C,X)	Hypergeometric F
IBETA(A,B,X)	Incomplete Beta Function
IGAM(A,X)	Incomplete Gamma
IGAMC(A,X)	Incomplete Gamma, Complement
LGNDR(L,M,X)	Legendre Polynomial PLM(X)
LNFACT(X)	Natural Logarithm Factorial
LNGAM(X)	Natural Logarithm Gamma
PFBI(K,N,P)	Binomial Probability (Probability $\geq K$ Occurrences in N trials, $P = \text{probability/trial}$)
PFCP(K,N)	Cumulative Poisson Probability (Probability 0 to K-1 Occurrences for Mean=N)
PFF(V1,V2,F)	F Probability (Probability Variance 1 $\geq$ Variance 2 for $F = f\text{-statistic}$,
V1=DOF1,V2=DOF2)	
PFST(T,V)	Student's t Probability (Probability $t < \text{Observed } t$ for $v = \text{DOF}$)
PFX2(X2,V)	Chi-Squared Probability (Probability Observed $C2 < C2$ for v DOF)
PFX2C(X2,V)	Chi-Squared Complement Probability (Probability Observed $C2 > C2$ for v DOF)
PSI(X)	Psi (Digamma) Function

Bessel Functions

J0(X) J1(X) JN(N,X)	Integer Order J Bessel, orders 0, 1, N
Y0(X) Y1(X) YN(N,X)	Integer Order Y Bessel, orders 0, 1, N
I0(X) I1(X) IN(N,X)	Integer Order I Modified Bessel, orders 0, 1, N
I0EIX(X) I1EIX(X)	$\text{EXP}(-X)$ * Integer Order I Modified Bessel, orders 0, 1
K0(X) K1(X) KN(N,X)	Integer Order K Modified Bessel, orders 0, 1, N
JNU(NU,X)	Fractional Order J Bessel

YNU(NU,X)	Fractional Order Y Bessel
INU(NU,X)	Fractional Order I Modified Bessel
KNU(NU,X)	Fractional Order K Modified Bessel
DJNU(NU,X)	Fractional Order J Bessel 1st Derivative
DYNU(NU,X)	Fractional Order Y Bessel 1st Derivative
DINU(NU,X)	Fractional Order I Modified Bessel 1st Derivative
DKNU(NU,X)	Fractional Order Modified K Bessel 1st Derivative
SPHJN(N,X)	Spherical Bessel j
SPHYN(N,X)	Spherical Bessel y
SPHIN(N,X)	Spherical Modified Bessel i
SPHKN(N,X)	Spherical Modified Bessel k
DSPHJN(N,X)	Spherical Bessel j 1st Derivative
DSPHYN(N,X)	Spherical Bessel y 1st Derivative
DSPHIN(N,X)	Spherical Modified Bessel i 1st Derivative
DSPHKN(N,X)	Spherical Modified Bessel k 1st Derivative
AIRYA(X)	Airy A Function
AIRYB(X)	Airy B Function
DAIRYA(X)	Airy A 1st Derivative
DAIRYB(X)	Airy B 1st Derivative

Derivatives

DX(Fn) - 1st Derivative of function Fn, $d(Fn)/dX$. Derivative is always with respect to X. UDF Example (1st Derivative of Gaussian):

```
F1=#A*EXP(-((X-#B)/#C)^2)
Y=DX(F1)
```

DX2(n) - 2nd Derivative of function #Fn, $d2(\#Fn)/dX2$. Derivative is always with respect to X. UDF Example (2nd Derivative of Gaussian):

```
F1=#A*EXP(-((X-#B)/#C)^2)
Y=DX2(F1)
```

Integrals

AI(n,st,end) and **AIP(n,st,end,prec)** - Automated Integration of #Fn from st to end with \$ as variable of integration; supports infinite limits and undefined bounds; AI() seeks 1E-8 precision, AIP() seeks prec fractional convergence. The AI() and AIP() functions first attempt to achieve the target precision with a successive step Gaussian Quadrature procedure. If this is unsuccessful, a Romberg procedure with a maximum of 131K-177K steps follows. For compatibility with previous versions, less automated integration procedures are also supported:

RI(n,st,end) - Romberg Integration

QI(n,st,end) - Gaussian Quadrature Integration, 24 step

QIN(n,st,end,nstep) - Gaussian Quadrature Integration, nstep steps (avl steps=4,6,8,10,12,16,20,24,32,40,48,64,80,128)

QIP(n,st,end,prec) - Gaussian Quadrature Integration, prec fractional convergence

UDF Example (Cumulative of Log-Normal):

```
UPPER=40.0
```

```

F1=LN($/#C)/#D
F2=EXP(-0.5*F1*F1)
F3=AIP(F2,X,UPPER,1e-4)
Y=#A+#B*F3

```

INF and -INF - Infinite Limits for Integration functions. You should always use the INF constant rather than some arbitrarily large number.

SUM(n,st,end,inc) - Sums #Fn with index \$ going from st to end with increment inc

SER(n,st,inc,lim) - Sums #Fn with index \$ beginning at st, incrementing with inc, until iteration's fractional contribution lim

PROD(n,st,end,inc) - Multiplies #Fn with index \$ going from st to end with increment inc

XY Data Table Constants And Functions

XMIN	Minimum X
XMAX	Maximum X
XRANGE	Max X - Min X
XMEAN	Mean X
XSTD	Std Deviation X
XMED	Median X
XATYMIN	X at Min Y
XATYMAX	X at Max Y
YMIN	Minimum Y
YMAX	Maximum Y
YRANGE	Max Y - Min Y
YMEAN	Mean Y
YSTD	Std Deviation Y
YMED	Median Y
YATXMIN	Y at Min X
YATXMAX	Y at Max X
XCTR	X at Peak Center (Peak Profiles Only)
XL50	X at Half-Maxima Left (Peak Profiles Only)
XR50	X at Half-Maxima Right (Peak Profiles Only)
XW50	X Width at Half-Maxima (Peak Profiles Only)
X50	X at Ymin+Yrange/2
X25	X at Ymin+Yrange/4
X75	X at Ymin+3*Yrange/4
XWTR	X transition width (X75-X25)
XWL	Wavelength (Waveform Profiles Only)
XPH, XPH2	Phase for Sine Wave and Sine2 Wave (Waveform Profiles Only)
XYCNT	Active Data Point Count
XYINDEX	Position within XY Data Table
NOISE(P)	Random Uniform P% Y Noise. A calculation Y=Y+NOISE(10) adds 10% uniform random noise.
GNOISE(P)	Random Gaussian P% Y Noise. A calculation Y=Y+GNOISE(10) adds 10% Gaussian random noise.

Non-Linear Functions

Peak Functions

ADC_(A0,A1,A2,A3,A4)
ADS_(A0,A1,A2,A3,A4)
ASYMLGSTC_(A0,A1,A2,A3)
BETA_(A0,A1,A2,A3,A4)
CHISQ_(A0,A1,A2,A3)
EQUILPK_(A0,A1,A2,A3,A4)
EMAXPOWPK_(A0,A1,A2)
EMG_(A0,A1,A2,A3)
ERFCPK_(A0,A1,A2)
ERROR_(A0,A1,A2,A3)
EXTRVAL_(A0,A1,A2)
EXTRVAL4T_(A0,A1,A2,A3)
EXTRVAL4F_(A0,A1,A2,A3)
FVAR_(A0,A1,A2,A3,A4)
GAMMA_(A0,A1,A2,A3)
GAUSS_(A0,A1,A2)
GAUSSLORXP_(A0,A1,A2,A3)
GAUSSMOD_(A0,A1,A2,A3)
GMG_(A0,A1,A2,A3)
INTMEDPK_(A0,A1,A2,A3)
INVGAMMA_(A0,A1,A2,A3)
LAPLACE_(A0,A1,A2)
LDRPK_(A0,A1,A2)
LGSTCPOWPK_(A0,A1,A2,A3)
LOGISTIC_(A0,A1,A2)
LOGNORM_(A0,A1,A2)
LOGNORM4_(A0,A1,A2,A4)
LORENTZ_(A0,A1,A2)
PEARSON4_(A0,A1,A2,A3,A4)
PEARSON7_(A0,A1,A2,A3)
PULSE_(A0,A1,A2)
PULSEPOW_(A0,A1,A2,A3)
PULSEWID2_(A0,A1,A2,A3)
SDC_(A0,A1,A2,A3)
SDS_(A0,A1,A2,A3)
STUDENT_(A0,A1,A2,A3)
WEIBULL_(A0,A1,A2,A3)

Asymmetric Double Gaussian Cumulative
Asymmetric Double Sigmoidal Peak
Asymmetric Logistic Peak
Beta Peak
Chi-Squared Distribution
Kinetic Intermediate w/Lag,Equilibrium
Exp to Max w/Power Term Peak
Exponentially Modified Gaussian
Complementary Error Function Peak
Error Distribution
Extreme Value
Extreme Value, 4-parameter tailed
Extreme Value, 4-parameter fronted
F-Variance Distribution
Gamma Peak
Gaussian
Gaussian-Lorentzian Cross Product
Modified Gaussian
Half-Gaussian Modified Gaussian
Kinetic Intermediate w/Lag
Inverted Gamma Distribution
Laplace or Double-Sided Exponential
Logistic Dose Response Derivative Peak
Logistic w/Power Term Peak
Logistic Peak
Log Normal
Log Normal, 4-parameter
Lorentzian
Pearson IV
Pearson VII
Pulse Peak
Pulse w/Power Term Peak
Pulse w/Second Width Peak
Symmetric Double Gaussian Cumulative Peak
Symmetric Double Sigmoidal Peak
Student's t Distribution
Weibull Peak

Transition Functions

ASYMSIG_(A0,A1,A2,A3)
ASYMSIGREV_(A0,A1,A2,A3)
CASCADE_(A0,A1,A2,A3)
EMGCUM_(A0,A1,A2,A3)
EXTRVALCUM_(A0,A1,A2)
GAUSSCUM_(A0,A1,A2)
IMPULSE_(A0,A1,A2,A3,A4)

Asymmetric Sigmoid
Asymmetric Sigmoid w/Reverse Asymmetry
Cascade Transition
Exponentially Modified Gaussian Cumulative
Extreme Value Cumulative
Gaussian Cumulative
Impulse Transition

LDR_(A0,A1,A2)
 LOGNORMCUM_(A0,A1,A2)
 LORENTZCUM_(A0,A1,A2)
 PULSECUM_(A0,A1,A2)
 PULSEPWCUM_(A0,A1,A2,A3)
 SDSCUM_(A0,A1,A2,A3)
 SIGMOID_(A0,A1,A2)
 WEIBULLCUM_(A0,A1,A2,A3)

Logistic Dose Response
 Log Normal Cumulative
 Lorentzian Cumulative
 Pulse Cumulative
 Pulse w/Power Term Cumulative
 Symmetric Double Sigmoid Cumulative
 Sigmoidal
 Weibull Cumulative

Kinetic Decay

DECAYPT5_(A0,A1)
 DECAY1_(A0,A1)
 DECAY2_(A0,A1)
 DECAY2HYP_(A0,A1)
 DECAY3_(A0,A1)
 DECAYN_(A0,A1,A2)
 DECAY1P2_(A0,A1,A2)
 DECAYAB\$C_(A0,A1,A2)
 DECAY1S1_(A0,A1,A2,A3)
 DECAY1S2_(A0,A1,A2,A3)
 DECAY2S2_(A0,A1,A2,A3)

Half Order Decay
 First Order Decay
 Second Order Decay
 Second Order Decay, Hyperbolic Form
 Third Order Decay
 Variable Order Decay
 Simultaneous First and Second Order Decay
 Decay of A for $A+B \rightarrow C$
 Two Independent First Order Decays
 Independent First and Second Order Decays
 Two Independent Second Order Decays

Kinetic Formation

FORMPT5_(A0,A1)
 FORM1_(A0,A1)
 FORM2_(A0,A1)
 FORM2HYP_(A0,A1)
 FORM3_(A0,A1)
 FORMN_(A0,A1,A2)
 FORM1P2_(A0,A1,A2)
 FORMSEQ_(A0,A1,A2)
 FORM1S1_(A0,A1,A2,A3)
 FORM1S2_(A0,A1,A2,A3)
 FORM2S2_(A0,A1,A2,A3)

Half Order Formation
 First Order Formation
 Second Order Formation
 Second Order Formation, Hyperbolic Form
 Third Order Formation
 Variable Order Formation
 Simultaneous First and Second Order Formation
 Sequential First Order Formation
 Two Independent First Order Formations
 Independent First and Second Order Formations
 Two Independent Second Order Formations

Kinetic Complex

EQUILRTA\$B_(A0,A1,A2)
 EQUILCNA\$B_(A0,A1,A2)
 EQUILA\$BC_(A0,A1,A2)
 EQUILAB\$CD_(A0,A1,A2)
 EQUILA\$B\$C_(A0,A1,A2,A3)
 INTMED_(A0,A1,A2)

Simple Equilibrium, Rate Form
 Simple Equilibrium, Concentration Form
 Complex Equilibrium $A=B+C$
 Complex Equilibrium $A+B=C+D$
 Intermediate Peak w/Equilibrium
 Intermediate Peak

Waveform Equations

SINE_(A0,A1,A2)
 SINE2_(A0,A1,A2)

Sine Wave
 Sine2 Wave

General Equations

POWER_(A0,A1)

Simple Power

EXP_(A0,A1)

Simple Exponential

PeakFit Functions

The PeakFit functions are a set of special peak, transition, and waveform functions built into PeakFit. These are made available in TableCurve 2D for cross-compatibility. The following abbreviations are used:

AMPL = Amplitude
CTR = Center
WID, WID1 = Peak or Transition Width
WID2 = Second Peak Width
WID3 = Third Peak Width
RATE, RATE1, RATE2 = Time Constant Terms
DIST = Chromatographic Distortion Parameter
PHASE = Waveform Phase
WL = Waveform Wavelength
DAMP = Waveform Dampening Parameter

The functions are as follows:

GAUSS(AMPL, CTR, WID)	Gaussian Peak
LORENTZ(AMPL,CTR,WID)	Lorentzian Peak
PEARSON(AMPL,CTR,WID1,WID2)	Pearson VII Peak
VOIGT(AMPL,CTR,WID1,WID2)	Voigt Peak
GAUSSA(AREA,CTR,WID)	Area Gaussian Peak
EMG(AREA,CTR,WID,RATE)	Exponentially-modified Gaussian Peak
GIDDING(AREA,CTR,WID)	Gidding Peak Equation
HVL(AREA,CTR,WID,DIST)	Haarhoff VanderLinde Peak
NLC(AREA,CTR,WID,DIST)	Non-Linear Chromatography Peak
LOGNORM(AMPL,CTR,WID)	Log Normal Peak
DBLEXP(AMPL1,RATE1,AMPL2,RATE2)	Double Exponential
DBLSIG(AMPL,CTR,WID1,WID2)	Symmetric Double Sigmoid Peak
DBLCUM(AMPL,CTR,WID1,WID2)	Symmetric Double Cumulative Peak
ADBLSIG(AMPL,CTR,WID1,WID2,WID3)	Asymmetric Double Sigmoid Peak
ADBLCUM(AMPL,CTR,WID1,WID2,WID3)	Asymmetric Double Cumulative Peak
SINE(AMPL,PHASE,WL)	Sine Wave
SINESQ(AMPL,PHASE,WL)	Sine Squared Wave
SINEDAMP(AMPL,PHASE,WL,DAMP)	Dampened Sine Wave
EXTRVAL(AMPL,CTR,WID)	Extreme Value Peak
GAMMA(AMPL,CTR,WID1,WID2)	Gamma Statistical Peak
WEIBULL(AMPL,CTR,WID1,WID2)	Weibull Statistical Peak
BETA(AMPL,CTR,WID1,WID2,WID3)	Beta Statistical Peak
LOGISTIC(AMPL,CTR,WID)	Logistic Peak
ERFCPK(AMPL,CTR,WID)	Error Function Peak
SIG(AMPL,CTR,WID)	Sigmoid Transition
CUM(AMPL,CTR,WID)	Gaussian Cumulative Transition
REVSIG(AMPL,CTR,WID)	Reverse Sigmoid Transition
REVCUM(AMPL,CTR,WID)	Reverse Cumulative Transition
LDR(AMPL,CTR,WID)	Logistic Dose Response Transition

SGLEXP(AMPL,RATE)

Single Exponential

Implicit UDF Models

Starting with v4.02, TableCurve 2D's UDFs can contain implicit functions. Implicit functions are of the form $f(x,y)=0$ where it is impossible to factor y to where it exists only on the left hand side of the equation.

The implicit UDF function is **IMPLICIT(Fn#,Ymin,Ymax,Tol)**

- **Fn#**, the number of the Fn to be implicitly solved for Y
- **Ymin**, the initial lower bracket for the solution of Y
- **Ymax**, the initial upper bracket for the solution of Y
- **Tol**, the fractional tolerance for solution convergence

If the initial bracketing for Y is unsuccessful, the initial Y limits are expanded first by 1x the Y range, then 10x the Y range, and finally by 100x the Y range. Zero is returned in the instance where bracketing or convergence fails.

The following UDF example is for the Michaelis-Menten equation for **enzyme kinetics** $Y=A0+A2*LN(A0)-A1*X-A2*LN(Y)$. Y is concentration, X is time, A0 is the concentration at time 0, A1 and A2 are the parameters of the underlying hyperbolic rate equation $dC/dt=A1*C/(A2+C)$:

```
F1=IF((Y.EQ.DATAERR).OR.(Y.LE.0),0,A0+A2*LN(A0)-A1*X-A2*LN(Y)-Y)
Y=IMPLICIT(F1,YMIN,YMAX,1E-8)
```

The following UDF example is for the basic **ligand binding** model $Y=((A0*A1/(1+A0*(10^X-Y*10^X))+A2)*(10^X-Y*10^X))/10^X$. Y is the Bound/Total ratio, X is the base 10 log of the Total, A0 is the affinity K, A1 is the binding capacity R, and A2 is the non-specific binding N:

```
F1=10^X
F2=((A0*A1/(1+A0*(F1-Y*F1))+A2)*(F1-Y*F1))/F1-Y
Y=IMPLICIT(F2,YMIN,YMAX,1E-8)
```

Note that the Y= is not a part of the expression to be solved. The Fn is implicitly solved for Y assuming the Fn evaluates to 0. If your implicit function is of the form $y=f(x,y)$ or $1=f(x,y)$, you must be sure to subtract Y or 1 from the Fn expression.

The file **MICHMENT.UDL** contains implicit and explicit solutions to the Michaelis-Menten equation. The UDF library, **LIGAND.UDL** contains six implicit models for a single ligand binding to 1, 2, and 3 sites, with and without non-specific binding. Also included are the two explicit solutions for the 1 site case.

Ligand UDF Library

TableCurve 2D ships with LIGAND.UDL, a UDF library containing six implicit ligand binding models. This library offers **implicit fits** for a single ligand with 1, 2, and 3 binding sites, both with and without non-specific binding. The library also contains the explicit solutions for the single site case.

These are the models developed by Feldman (Henry A. Feldman, Mathematical Theory of Complex Ligand Binding Systems at Equilibrium: Some Methods for Parameter Fitting, Analytical Biochemistry, vol. 48, p317-338, 1972), and offered in the NIH Ligand software by Munson (Peter J. Munson and David Rodbard, LIGAND: A Versatile Computerized Approach for Characterization of Ligand Binding Systems, Analytical Biochemistry, vol. 107, p220-239, 1980).

The first and second UDFs are for the single site model. Y is the Bound/Total ratio, X is the base 10 log of the Total, A0 is the affinity K, A1 is the binding capacity R, and A2 is the non-specific binding N for the first UDF:

```
F1=10^X
F2=((A0*A1/(1+A0*(F1-Y*F1))+A2)*(F1-Y*F1))/F1-Y
Y=IMPLICIT(F2,YMIN,YMAX,1E-8)
```

The third and fourth UDFs are for the two site model. Y is the Bound/Total ratio, X is the base 10 log of the Total, A0 is the affinity K1, A1 is the binding capacity R1, A2 is the affinity K2, A3 is the binding capacity R2, and A4 is the non-specific binding N for the third UDF:

```
F1=10^X
F2=((A0*A1/(1+A0*(F1-Y*F1))+A2*A3/(1+A2*(F1-Y*F1))+A4*(F1-Y*F1))/F1-Y
Y=IMPLICIT(F2,YMIN,YMAX,1E-8)
```

The fifth and sixth UDFs are for the three site model. Y is the Bound/Total ratio, X is the base 10 log of the Total, A0 is the affinity K1, A1 is the binding capacity R1, A2 is the affinity K2, A3 is the binding capacity R2, A4 is the affinity K3, A5 is the binding capacity R3, and A6 is the non-specific binding N for the fifth UDF:

```
F1=10^X
F2=((A0*A1/(1+A0*(F1-Y*F1))+A2*A3/(1+A2*(F1-Y*F1))+A4*A5/(1+A4*(F1-Y*F1))+A6*(F1-Y*F1))/F1-Y
Y=IMPLICIT(F2,YMIN,YMAX,1E-8)
```

The seventh UDF is identical to the first except that an explicit solution is used for an appreciably faster fit:

```
F1=10^X
Y=(A0*A1+A2+2*A0*A2*F1+1+A0*F1-SQRT(2*A2+2*A0*F1+1+A0*A0*(-
2*A1*F1+A1*A1+F1*F1)+2*A0*A1+2*A0*A1*A2+A2*A2+2*A0*A2*F1))/(2*A0*F1*(A2+1))
```

The eighth UDF is identical to the second except that again an explicit solution is utilized:

```
F1=10^X
Y=(A0*A1+1+A0*F1-SQRT(2*A0*F1+1+A0*A0*(-2*A1*F1+A1*A1+F1*F1)+2*A0*A1))/(2*A0*F1)
```

The formulas used for the estimates are general ones. Adjustments should be made with the **UDF Adjust** procedure to insure successful fitting. This is especially true of the multiple site models.

For these UDFs, the models fit B/T vs. base 10 log of T. If your data consists of B/T vs. T, you can either apply an X=LOG(X) calculation to the data prior to fitting, or you can change the F1=10^X statements to F1=X within the individual UDFs.

Michaelis-Menten UDL Library

TableCurve 2D furnishes the **user-defined function** library MICHMENT.UDL. The Michaelis-Menten equation, often used in enzyme **kinetics**, is based upon a hyperbolic rate equation: $dC/dt = A1 \cdot C / (A2 + C)$. The solution, $C(t)$, involves an implicit form: $Y = A0 + A2 \cdot \ln(A0) - A1 \cdot X - A2 \cdot \ln(Y)$. Y is concentration, X is time, $A0$ is the concentration at time 0, and $A1$ and $A2$ are the parameters of the underlying rate equation.

The first UDF solves this model implicitly:

```
F1=IF((Y.EQ.DATAERR).OR.(Y.LE.0),0,A0+A2*LN(A0)-A1*X-A2*LN(Y)-Y)
Y=IMPLICIT(F1,YMIN,YMAX,1E-8)
```

The second UDF offers an explicit solution via Lambert's W function:

```
Y=OMEGA((1/A2)*EXP(-(-A0-A2*LN(A0)+X*A1)*(1/A2)))*A2
```

The UDFs use the following formula for initial estimates:

```
A0=YMAX
A1=YMAX*(LN(3)-LN(2))/(2*X50*LN(3)+2*XWTR*LN(2))
A2=-YMAX*(X50+XWTR)/(2*X50*LN(3)+2*XWTR*LN(2))
```

where,

YMAX=maximum Y, should be value at time $X=0$ where data begins

X50 = X at 50% of Y range

XWTR = X at 75% Y range - X at 25% Y range

The initial estimates will be accurately determined, provided sufficient data exists for TableCurve 2D to accurately map a transition center and width.

Since all active UDFs are automatically fitted in all processing options, the **Curve Fit Kinetics Equations** can be used to fit this model and the standard kinetics models in a single step.

Last Session User Functions

The last session UDFs are automatically saved as a UDF library. The configuration file [TC2D2V4.CFG](#), is actually this UDF library. This file is automatically written upon exit, but an existing file is only overwritten when a UDF has been installed in the current session. When this [Last Session User Functions](#) item is selected, these UDFs are loaded, validated, and compiled.

Curve-Fit Preferences

This option is used to specify the fitting controls that are used for all of TableCurve 2D's Curve-Fit options except the [Curve-Fit Peak Functions](#), the [Curve-Fit Transition Functions](#), and the [Curve-Fit Kinetic Equations](#) which have their own separate non-linear controls.

Linear fitting controls include the default fitting procedure, the term significance threshold, the maximum order of equation permitted, forcing a zero intercept, using base 10 rather than natural logarithms, and disabling the degree of freedom normally assigned to the error. Non-linear controls include the maximum allowable number of iterations, the convergence precision, the minimization criteria, and intercept options. You can save a custom fit configuration to disk for recall at any future time.

Linear Fitting Procedure

Unless you have specific matrix methods expertise and wish to select a specific procedure for fitting all equations, the real decision here is whether to use the [Fast Std](#) option or the [Fast SVD](#) procedure. Both use a mix of methods to minimize processing time. The difference is whether higher order polynomials and rationals are fitted by a conventional method (Gaussian Elimination) or by Singular Value Decomposition (SVD).

Higher order polynomials and rationals tend to produce ill-conditioned design matrices. When solved by conventional algorithms, such as Gaussian Elimination, the coefficients will often span many orders of magnitude and they will tend to be very large in magnitude and alternating in sign. Changing a seemingly insignificant digit in only one of the coefficients could drive the function far from the data.

With SVD, the solution will be more stable. The coefficients are driven toward zero rather than infinity by the ill-conditioning, and the result is that the coefficients will span fewer orders of magnitude and they will tend to be smaller in magnitude. A small change in a single coefficient is less likely to drive the function away from the data and limited extrapolations may even be possible. The disadvantages of SVD are that its results will likely produce a poorer goodness-of-fit statistic and the method requires an order of magnitude greater fitting time. In some cases, the zeroing that occurs within SVD can cause the fit to fail entirely.

The [Fast SVD](#) method assumes you wish this additional SVD stability in higher order polynomials and rationals. The [Fast Std](#) method assumes you wish the conventional polynomial or rational solution.

If you are using the [Fast SVD](#) method and not getting the results you expect for higher order polynomials and rationals, you may want to select the [Fast Std](#), or the specific [Gauss-Jordan](#) or [LU-Decomposition](#) methods.

Note that selecting a specific method will slightly increase the fitting time and will typically reveal differences only with the higher order polynomial and rational functions. Note also that TableCurve 2D exclusively uses extended double precision (80 bit) floating point math. It is possible that no method selected will match the polynomial coefficients produced by another program which uses double precision (64 bit) or single precision (32 bit) math.

Highest Term Count to Be Fitted

To change the maximum coefficient count permitted when fitting linear equations, enter any value between 2 and 21. The default is 21, which permits all of TableCurve's equations. Note that the maximum term count applies to all linear fits, including the built-in equation sets.

It is recommended that you reset this count back to 21 when you have completed the limited term fitting. This prevents the unfortunate case in which a low term count preference is subsequently forgotten and a

succession of All Equations fits are made only to later discover that the higher term equations were not included.

The maximum term count may be overridden to insure a degree of freedom of one, (or zero, if the **No DOF For Error** option is checked).

The **Arrange by Parameter Count** option in the Review's Equation List sorts the candidate equations by term count, and is probably a better way to manage the inspection of simpler equations.

Term Significant Digit Threshold

The default Term Significance is 9. This means that a given term in a linear equation must make at least a 1e-9 (0.0000001%) contribution to the average Y value when evaluations are made at Xmin, Xmean, and Xmax. A term significance of 9 will only rarely discard an equation. You may enter any value for term significance from 1 to 15. At a low value such as 2 (1e-2,1%), a considerable number of equations may be excluded. Although there is no direct correlation, you may wish to set the term significance at a value slightly above the number of significant figures in your data.

Force Zero Intercept on Standard Equations

When this box is checked, all of TableCurve's standard equations, Eqns 1-105, 1001-1630, 2001-3330, and 4001-5140, will be forced through the origin. This option reduces the design matrix of the full equation so as to produce a 0.0 value for the intercept term. You should use this option only when it is absolutely necessary for a given equation to pass through the origin. Note that TableCurve's standard equations with log and inverse terms are less likely to produce anything of value when this option is selected. With this option active, the **a** coefficient of the affected equations will appear as 0.0 and will have no standard error.

Log Equations Ln=Base 10 Log Rather than Natural Log

When this box is checked, all of TableCurve's linear equations containing the natural log function become common or base 10 log functions. Note that while this option may benefit the simple equations relative to subsequently plotting them with TableCurve's base 10 common log graphs, the various log polynomials and rationals are likely to produce poorer fits since a wider dynamic range is being raised to the higher powers.

No DOF For Error (For Fitting Error-Free Data)

This option only affects the equations fitted when there are fewer than 22 points in the data set. When this box is checked, TableCurve does not reserve a degree of freedom for the error of fit. This means it will be possible to fit a polynomial with five coefficients to five data points. Normally, functions with five or more coefficients would not be fitted when only five data points are present. This option should be used only when fitting generated or handbook data known to have negligible error.

Non-Linear Minimization

In addition to standard least-squares minimization, TableCurve 2D's non-linear engine is capable of three different **Robust Estimations**. These minimizations are sometimes referred to as maximum likelihood or m-estimate fitting.

If your data span a large number of orders of magnitude in the Y variable and the low-valued Y points are not factoring into the fit, a robust fit will remedy this problem. This may be a superior solution to fitting the LN(Y) rather than Y or seeking to weight the data so that these low-valued Y points can factor into a least-squares solution.

The other instance in which robust fitting is recommended is if it is known that there are significant outliers within the data. Robust estimation will minimize the impact of outliers.

Least-squares corresponds to a Gaussian maximum likelihood distribution of errors. All of the **robust minimizations** correspond with maximum likelihood probability distributions significantly less compact than the Gaussian. These wider tails mean that the errors associated with outliers are expected. When outliers are suspected or likely, the Lorentzian minimization is highly recommended. Although TableCurve's Levenburg-Marquardt non-linear engine is used for all of these minimizations, you will generally find that a higher number of iterations will be required for a robust fit as compared to least squares.

Maximum Iterations

The default number of iterations for a non-linear equation is 200. You may enter any value between 2 and 9999. There is seldom much to gain beyond 200 iterations unless you are working with a very complex model or one whose parametrization evidences a very slow convergence.

Converge to Significant Digits in r^2

The default non-linear convergence precision is 6. This means that the r^2 coefficient of determination must be unchanging in the sixth decimal place for five consecutive iterations to signal convergence. You may enter any value between 3 and 15. Note that often only a few additional iterations are needed to converge to a high precision. A convergence precision of 12 may thus require very little additional fitting time than the default of 6.

Intercept Options

Most of TableCurve's non-linear equations have both an intercept and zero-intercept form. There are three intercept options where you can fit only those non-linear equations with intercepts, only those with zero-intercepts, or both versions.

Saving Fit Preferences

Use the **Save** item to save the current preferences to disk. The default file extension is [FTP]. These are binary files that can only be produced within the program. The current preferences are always saved automatically across sessions. Save preferences to disk if you plan to regularly use more than one fit configuration in your work. Curve-Fit Preference files can be recalled at anytime using the **Read** item.

Reset

The **Reset** button restores TableCurve's default curve-fit preferences.

Background Thread Processing

Multitasking operating systems offer the means to simultaneously carry out different computational procedures. When available, TableCurve 2D makes full use of this multithreading environment. One of the principal applications for multithreading is to have TableCurve 2D's automated fitting occurring on one thread, while other program operations are occurring.

Preselected Equations for Thread Fitting

When background thread fitting is enabled, TableCurve 2D initiates a preselected type of fitting the moment data are loaded or revised by a procedure such as smoothing or FFT editing. Thus, while in light CPU tasks such as entering titles or viewing data, TableCurve 2D can be processing the data in the background, often having the Review ready by the time you return to the main menu. This option is primarily of use when fitting many similar data sets or if you use one specific fitting option for your fitting.

Thread Fitting Options

The [Background Thread Processing](#) dialog offers the following options:

- None - Use this option to disable background thread fitting
- All Equations
- Linear Equations
- Simple Equations
- Robust Straight Line
- Poly-Ratl Equations
- Waveform Equations
- Current Custom Equation Set
- Current User Functions
- Peak Functions - Current
- Peak Functions - All
- Peak Functions - AI Expert
- Transition Functions - Current
- Transition Functions - All
- Transition Functions - AI Expert
- Kinetic Equations - Current
- Kinetic Equations - All
- Kinetic Equations - AI Expert

The fitting that occurs via a background thread is numerically identical to that which would occur if the fit option had been selected from the Process menu using the current settings for the applicable fit controls. Since the background thread is a pure numeric process without the overhead of a messaging queue and a graphical update, its processing times are very rapid when the CPU demand is light.

Thread Fitting in Progress Indicators



When you are back at the menu of TableCurve 2D's main window, the light bulb button in the tool bar reflects the status of the background thread fitting. Unlike the other tool bar buttons, this light bulb button is purely informational. When fitting is underway, but not yet complete, this bulb is lit. Once the fitting is completed, the bulb goes out. If the fitting is already completed prior to your return to the main menu, this bulb will not be lit, but the two buttons for starting the Review will be active, indicating a successful

background fit.

The status window graph also reflects the current state of the background thread fitting. If the fitting is complete when you return to the main menu, this graph will plot the highest ranked equation of those added during the fitting. If the background thread fitting is not complete, the graph will first show the raw data and then change to the highest ranked equation once the fitting has been completed.

When the [Audible Signal when Processing Complete](#) option is checked, TableCurve 2D produces an audible signal on the computer's speaker when the background thread fitting is complete.

Automatic Start of Review

The [Automatically Start Review when Processing Complete](#) option, when checked, results in TableCurve 2D automatically launching the Review when the background fitting is complete and you have returned to the main menu. If the fitting is complete when you return to the main menu, TableCurve 2D will skip the status window update and proceed to directly open the [Curve-Fit Review](#) with the option last used to begin such. If the fitting is still in progress when you return to the main menu, the status window graph will initially reflect the raw data and the light bulb button in the tool bar will be lit. Once the fitting is completed, the light bulb button will go out and the Review will be automatically started.

Cancellation of Background Thread Fitting

Any of the Process menu's fitting options cancel the background thread fitting, even if you select the very fitting option that is underway. Any operation which alters the data table in any way will also cancel the background thread fitting, although the fitting will be automatically resumed when it is appropriate. For example, the background thread fitting is discarded when you enter the smoothing procedure, and it is reinitiated once you have accepted the result of the smoothing.

Some Things to Do While Background Thread Fitting is in Progress

Here are some suggestions as to what you can do while TableCurve 2D is fitting data on a background thread:

- [Enter accurate titles](#) either following the import or edit procedure, or using the Titles option in the Table menu.
- [View the data graphically](#) to inspect for outliers. The background fit is only halted and reinitiated if you section a range of data or toggle a point on or off.
- [Open the TableCurve Editor](#) or [ASCII Editor](#) to view imported data numerically. The background fit is only halted and reinitiated if you alter the data.
- [Save the data table](#) if it is appropriate.
- Explore TableCurve 2D features via the help system.
- Work within any other application.

TCEX.DLL Functions

TableCurve 2D supports the inclusion of external C-language or Fortran language functions within its equation set. These are known as TCEX.DLL functions, since the implementation requires these functions exist within a DLL named TCEX, and that a specific format be followed.

TableCurve 2D provide a C-language template TCEX.C for use with Microsoft Visual C++ and Borland C++. The Fortran language template is in TCEX.F90 and requires Microsoft/DEC Fortran Powerstation. These two templates contain extensive instructions for implementing the functions and preparing the DLL.

TCEX.DLL must be present in the TableCurve 2D distribution directory at the time the program is loaded. If it is detected, the various functions contained within it are added to TC2D's function set. These equations will begin at Equation # 9001 and can range up to Equation #9100. You may thus have up to 100 TCEX.DLL functions.

The Process menu will contain a [TCEX.DLL Functions](#) item only if one or more functions were successfully detected at the time of startup.

You have an option of disabling all of the functions, enabling all of them, or selecting only those desired for fitting. Use the standard Windows procedure for selecting multiple items in the selection list (holding mouse down and moving through list to highlight multiple items, using Shift key with second selection to create a band between this and the first selection, or using the Control key to append individual selections).

Fit Controls

The non-linear fitting controls applicable to TCEX.DLL functions can be accessed via the [Options](#) button. This opens the [Curve Fit Preferences](#) option.

Fitting the TCEX.DLL functions

All of TableCurve 2D's fitting options automatically include the TCEX.DLL functions that are currently enabled.

Note that if the TCEX.DLL functions are fit with the Peak, Transition, or Kinetic models, the custom preferences for these fitting options will override the general preferences.

 When no UDFs are currently loaded, the [Curve Fit User Functions](#) option can be used to fit only the TCEX.DLL equations.

Review Menu



TableCurve 2D offers two default configurations for beginning the Curve-Fit Review. The **Graph Start** option offers a screen where the curve-fit graph and equation list are simultaneously displayed. The **List Start** option presents a large equation list that is closed after each selection. The Review is designed to be refined as it is used, the layout and functionality taking shape in accord with your preferences. TableCurve 2D saves two separate Review desktops across sessions, one for each of these start options.

The Review consists of the following elements: the **Curve-Fit Graph**, the **Equation List**, the **Residuals Graph**, the **Numeric Summary**, the **Data Summary**, the **Precision Summary**, and the **Evaluation**. The Curve-Fit Graph is always displayed. You may display any or all of the other items simultaneously. Each of the elements offers its own particular file and printed output.

Review Menu



Graph Start



List Start

Main Window (Curve-Fit Graph)

Curve-Fit Graph

Secondary Windows



Eqn. List Equation List



Resid. Residuals Graph



Numeric Numeric Summary



Data Data Summary



12345 Prec. Precision Summary



Eval. Evaluation

External Export Options



Excel Export



Language Code Export



SigmaPlot Export

Lotus 123 Export

Harvard Graphics Export

ASCII Export Options

ASCII Export

Parameters Export

Covariance Matrix Export

TableCurve 2D UDF Export

UDF Generation

Other Review Topics



Mark Excluded Points for Refit



Confidence and Prediction Intervals



Curve-Fit Graph Format



Equation References

Preferences

Goodness of Fit Statistics



Graph Start

The **Graph Start** option offers an initial Review screen which simultaneously displays the curve-fit graph and equation list. The Review desktop is refined as you determine your particular preferences.

The Review consists of the following elements: the **Curve-Fit Graph**, the **Equation List**, the **Residuals Graph**, the **Numeric Summary**, the **Data Summary**, the **Precision Summary**, and the **Evaluation**. Each of these windows has its own menu. The Curve-Fit Graph is always displayed. You may display any or all of the other items simultaneously. Each of the elements offers its own particular file and printed output.

List Start

The [List Start](#) option offers an initial Review screen which emphasizes the equation list. This list closes after each equation selection allowing the largest possible curve-fit graph. The Review desktop is refined as you determine your particular preferences.

The Review consists of the following elements: the [Curve-Fit Graph](#), the [Equation List](#), the [Residuals Graph](#), the [Numeric Summary](#), the [Data Summary](#), the [Precision Summary](#), and the [Evaluation](#). Each of these windows has its own menu. The Curve-Fit Graph is always displayed. You may display any or all of the other items simultaneously. Each of the elements offers its own particular file and printed output.

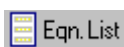
Curve-Fit Graph

The [Curve-Fit Graph](#) is the central point in the TableCurve 2D Review.

The curve-fit will be displayed in a [TableCurve 2D Graph](#) that plots the X-Y data along with the fitted equation, and optionally with confidence and/or prediction intervals, error bars, and fit references. The graph can be scaled and formatted, a variety of file output options are available, the graph can be printed, residuals, derivatives and cumulative areas of the fit equation can be displayed in a Y2 axis, and a full numeric evaluation can be made of the selected equation.

The Curve-Fit graph is also the control point for toggling on and off the the [Equation List](#), the [Residuals Graph](#), the [Numeric Summary](#), the [Data Summary](#), the [Precision Summary](#), and the [Evaluation](#). The Curve-Fit graph cannot be closed during the course of the Review. You can move and size this window as you wish, although you cannot hide the menu or control buttons. The position and size of the Curve-Fit graph are saved across sessions.

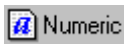
Toggling Other Windows On and Off



The [Eqn. List](#) button is used to toggle the [Equation List](#) window.



The [Resid.](#) button is used to toggle the [Residuals Graph](#) window.



The [Numeric](#) button is used to toggle the [Numeric Summary](#) window.



The [Data](#) button is used to toggle the [Data Summary](#) window.



The [Prec.](#) button is used to toggle the [Precision Summary](#) window.



The [Eval.](#) button is used to toggle the [Evaluation](#) window.



Note that these buttons are smaller if the video resolution is 800x600.

Automatically Tiling Windows



The [Curve-Fit Graph](#) and [Equation List](#) are automatically opened and positioned.



The [Curve-Fit Graph](#), [Equation List](#), and [Numeric Summary](#) are automatically opened and positioned.



The [Curve-Fit Graph](#), [Equation List](#), and [Data Summary](#) are automatically opened and positioned.



The [Curve-Fit Graph](#), [Equation List](#), [Numeric Summary](#), and [Residuals](#) are automatically opened and positioned.



The [Curve-Fit Graph](#), [Equation List](#), [Numeric Summary](#), and [Evaluation](#) are automatically opened and positioned.



The [Curve-Fit Graph](#), [Equation List](#), [Residuals](#), and [Evaluation](#) are automatically opened and positioned.

Note that all other windows are closed when selecting one of these options and that current windows settings are replaced. Also, 1024x768 or greater video resolution is recommended for the options that simultaneously display four different windows.

Y2 Axis Content



The [Set Y2 Axis Content](#) button is used to assign an optional Y2 axis. This can contain the residuals, the first and/or second derivative of the fitted curve, or the cumulative area of the fit equation.



The **Toggle Residuals on Y2 Axis** button offers a fast way to toggle the **Y2 Axis Residuals** on and off.



The **Toggle First Derivative on Y2 Axis** button offers a fast way to toggle the **Y2 First Derivatives** on and off.

For built-in curve-fit equations, both analytic and numeric derivatives are available. The type is specified in the **Preferences** option in the Curve-Fit graph's File menu. This is also available from the main **Edit** menu.

Zooming Out Curve-Fit Graph

The **Zoom-Out** scroll-bar is used to select one of 20 automatic zoom-out levels. The zoom-out is part of the graph's automatic scaling. The zoom-out will zoom linear axes in a linear fashion, and logarithmic axes by alternately adding a decade to the top or bottom of the range for each level of zoom-out.

Fit References



The **Equation References** button is used for most options that affect the installation and removal of curve-fit references. TableCurve 2D supports up to four simultaneously displayed equation references. These can be from the current equation set or imported from prior fits.



The **Toggle Display of Equation References** turns on and off the graphical display of the references.



The **Add Current Equation as Reference** button offers the means to install the current fit as a reference with a single button click.

Equation Scan and Search



This button selects the top-ranked equation in the **Equation List**.



This button scans up one equation in the **Equation List**.



This button scans down one equation in the **Equation List**.



The **Search for Specific Equation** button is used to directly select the TableCurve 2D equation desired. You must enter the equation number sought. If the equation is not in the **Equation List** for any reason, you will be shown a message that explains why the equation is not available. This is the fastest way to find a TableCurve 2D equation when you know in advance the equation you seek.

Using Mouse to Scroll Equations

When the Curve-Fit graph is the active window, you can use the rotating wheel on a wheel mouse to control movement through the **Equation List**. Rotate the wheel forward (away from you) to advance up the equation list and back (toward you) to step down the equation list. You can also single click on the middle button of a three-button mouse (or depress the wheel of a wheel mouse) to advance or go back one equation, depending on the current direction through the list.

The **Equation List** window can be toggled on and off by double clicking the middle button or wheel.

Mark and Refit



The **Mark Excluded Points for Refit** button is used to automatically mark points that will be toggled in state upon a refit. It is typically used to mark outliers for exclusion prior to refitting. Points can also be manually marked for subsequent exclusion/inclusion in fitting by left clicking on the data point in the Curve-Fit graph.



The **Refit** option will reflect the type of fit that was made prior to entering into the Review. When a refit is initiated, all marked active points are made inactive while marked inactive points are made active. You can go back and forth between including and

excluding a set of marked points by successive refits.

Common Export Options

 The **Copy Data to Clipboard** button is used to copy the current curve-fit in a **WK1 format** to the clipboard for pasting into Excel, SigmaPlot, Lotus, Quattro, and other WK1 compatible applications. The WK1 can be saved to disk using the **Save Lotus 123** option in the File menu of the Curve-Fit graph.

 The **Export to SigmaPlot** button is used to export the current curve-fit to one of four **SigmaPlot** graph file formats. This can also be done by selecting the **Save SigmaPlot** option in the File menu of the Curve-fit graph.

 The **Code Generation** button is used to generate C, FORTRAN, BASIC, or PASCAL **language code** for the current equation. This can also be done using the **Code Generation** option in the File menu of the Curve-fit graph.

 The **MS Excel Export** button is used to export the current curve-fit to **Excel**. This can also be done using the **Save Excel** option in the File menu of the Curve-fit graph.

 The **MS Word/RTF File Export** button is used to write a report containing one of more of the following elements to an **MS Word or RTF** file: current curve-fit graph, residuals graph, numeric summary, data summary, precision summary, evaluation, language code, equation list. This option can also be invoked by using the **Save MS Word/RTF** option in the File menu of the Curve-fit graph.

 The **Production Facility** button is used to invoke the **automated processing** of similar data sets using the currently selected curve-fit equation. All data sets in the automation are fitted to this one model.

TableCurve 2D Graph Options

The following curve-fit elements are accessed directly from the **TableCurve 2D Graph's** toolbar:

-  Opens **Confidence Intervals** dialog.
-  Toggles **Confidence/Prediction Intervals**.
-  Opens **Error Bars** dialog (Replicate information present)
-  Toggles **Error Bars** (Replicate information present)
-  Display actual **Residuals** (Residuals in Y2 axis)
-  Display **Residuals as % of Y**(Residuals in Y2 axis)
-  Display **Residuals as fraction of SE** (Residuals in Y2 axis)

File Menu-File Export Options

Report information can be written:

- to an **MS Word/RTF** file

You may export the curve-fit information:

- to a **Lotus 123 WK3** worksheet file
- to an **Excel XLS** worksheet
- to a **SigmaPlot Windows SP5** graph file
- to a **Harvard Graphics CHT** file.

The ASCII export options include saving:

- the **Generated Curve**
- the **Parameters**

- the **Covariance Matrix**

TableCurve 2D can also generate:

- **Language Code** for the curve-fit equation in C, PASCAL, BASIC, or FORTRAN
- TableCurve 2D **User-Defined Functions** from the current equation

File Menu-Preferences

The **Preferences** option is used to:

- set the numeric precision in graph titles
- specify right filling numeric values with zero
- specify a compact notational format for non-linear equations
- using bitmaps for rendering graphs
- specifying Rich Text Format (RTF) text display windows
- specifying analytic derivatives for all curve-fit equation derivative computations

Note that this option is identical to the **Preferences** option in the **Edit menu**.

File Menu-Printing



To **Print** the graph you can use the **Print** menu item or the **Print** button in the graph's toolbar. You can set up the desired printer with the **Printer Setup** menu item, or the **Setup** button in the **Print** dialog. In the Review, it is possible to print a half-page numeric summary below the standard half-page curve-fit graph.

Reference Menu

The **Equation References** item is used for the addition and removal of equation references. An equation reference overlays the current curve-fit. References are usually used for graphical comparison with other curve-fits in the current equation set, or with fits saved from a prior analysis. Up to four references can be active at a given time.



Note that there is also an **Equation References** button in the Curve-Fit graph's control panel.

The following operations can also be initiated directly using Reference menu:

- The **Add Current Equation as Reference** menu item is a convenient way to immediately install the current fit as a reference.



Note that there is also an **Add Current Equation as Reference** button in the Curve-Fit graph's control panel.

- The **Save Current Equation as Reference** menu item saves the current fit to disk as an equation reference.
- The **Import Reference Library** menu item is a quick way to import a set of previous saved references.
- The **Save Current Set of References as Library** menu item can be used to immediately save the current set of installed references to disk for later import.
- The **Display References** menu item turns on and off the graphical display of the references.



Note that there is also a **Toggle Display of Equation References** button in the Curve-Fit graph's control panel.

Scan Menu



The **First Equation**, **Next Higher Rank**, and **Next Lower Rank** items duplicate the VCR buttons

in the control panel and are likewise used to select the highest ranked fit, the next higher ranked fit, and the next lower ranked fit in the [Equation List](#). The [Last Equation](#) item selects the worst ranked equation in the set.



The [Search for Specific Equation](#) item is used to directly select the TableCurve 2D equation desired. You must enter the TableCurve 2D equation number. If the equation is not in the [Equation List](#) for any reason, you will be shown a message that explains why the equation is not available. This is the fastest way to find a TableCurve 2D equation when you know in advance which equation it is you seek.

List Menu

This menu duplicates the content items in the [Equation List](#) window's menu.

Window Menu

The [Curve-Fit Graph](#) item will simply make the Curve-Fit graph the current window since this graph is always displayed in the course of the Review. The selection of [Residuals Graph](#), [Equation List](#), [Numeric Summary](#), [Data Summary](#), [Precision Summary](#), or [Evaluation](#) will display the window if it is not already displayed, and then it will give this window the current focus.

The six [Tile](#) items are used to automatically open and place various windows. These options are also available in the Curve-Fit graph's control panel:



The [Tile Fit Graph/Equation List](#) item automatically opens and positions the [Curve-Fit Graph](#) and [Equation List](#).



The [Tile Fit Graph/Equation List/Numeric](#) item automatically opens and positions the [Curve-Fit Graph](#), [Equation List](#), and [Numeric Summary](#).



The [Tile Fit Graph/Equation List/Data](#) item automatically opens and positions the [Curve-Fit Graph](#), [Equation List](#), and [Data Summary](#).



The [Tile Fit Graph/Equation List/Numeric/Residuals](#) item automatically opens and positions the [Curve-Fit Graph](#), [Equation List](#), [Numeric Summary](#), and [Residuals](#).



The [Tile Fit Graph/Equation List/Numeric/Evaluation](#) item automatically opens and positions the [Curve-Fit Graph](#), [Equation List](#), [Numeric Summary](#), and [Evaluation](#).



The [Tile Fit Graph/Equation List/Residuals/Evaluation](#) item automatically opens and positions the [Curve-Fit Graph](#), [Equation List](#), [Residuals](#), and [Evaluation](#).

The [Save State of Review Windows](#) option saves a file that contains the information for the current state of the Review windows. This includes which windows are currently shown and their size and positions. These are binary files with a default WIN extension.

The [Import State of Review Windows](#) option reads a WIN file and opens the appropriate windows in their previously saved sizes and positions.



Curve-Fit Graph Format

The Curve-Fit graph can optionally have a Y2 axis. This is set using the [Set Y2 Axis Content](#) button in the [Curve-Fit Graph's](#) control panel.

Format Options

- Curve-Fit, Data in Y1
- Curve-Fit, Data in Y1; Residuals in Y2
- Curve-Fit, Data in Y1; First Derivative in Y2
- Curve-Fit, Data in Y1; Second Derivative in Y2
- Curve-Fit, Data in Y1; First and Second Derivatives in Y2
- Curve-Fit, Data in Y1; Cumulative Area in Y2

Derivatives and Cumulative Area

The Y2 axis can also optionally contain the first and/or second derivative of the fitted curve, or the cumulative area of the fit equation. The cumulative area always uses the minimum X of the data table as the lower limit of integration, regardless of the graph's minimum X value.

Residuals

To observe a direct correspondence between the fit and the residuals, the [residuals](#) can be plotted on the Y2 axis. Since the X axis must be common, only basic, percent, and standardized residual graphs will be available. These buttons will be added to the Curve-Fit graph's [toolbar](#) when the Residuals are plotted on the Y2 axis.



Display actual [Residuals](#).



Display [Residuals as % of Y](#).



Display [Residuals as fraction of SE](#) (standardized).

ASCII Export

You can export generated **Curve-Fit** information to an ASCII file. The file is created via the **Save ASCII** option in the File menu of the Curve-Fit Graph. You can generate the data with or without confidence and prediction intervals.

Structure of Exported ASCII File

- The **X,Y with Confidence Limits** item writes generated information based upon X values determined by the values in the **Start**, **Incr**, and **End** fields. This generated section will contain the predicted Y value as well as confidence and prediction intervals. The initial values in the X generation will produce a block of 100 points. The confidence value used will be the value currently set in the **Intervals** option.
- The **X,Y Only** option writes only the X,Y information defining the curve-fit in a two-column format.
- You can choose either **Space**, **Comma**, or **Tab** delimiters.
- You can also set the **Precision** from 3 to 18 significant figures.

After the file is written, you are presented with an **Exported ASCII File Display**.

Note that it is also possible to generate a columnar XY ASCII file from the Curve-Fit graph's **Evaluation** option. XY ASCII files generated in the Evaluation write the XY data in columnar format with 17 digits of precision and space delimiters, and can optionally include two additional columns with either confidence or prediction limits. If you need to generate XY data based upon X values contained within a file, you must use the Evaluation option.

Language Code Export

The **Code Generation** option in the control panel or File menu of the **Curve-Fit Graph** is used to write generated code for the current curve-fit or AR model equation. Code can be generated for all of TableCurve 2D's built-in equations and for all **AR model** fits.

Supported Languages

You can export any TableCurve 2D built-in **Curve-Fit** equation or **AR model** to program code in the following languages:

- C (double)
- C (long double)
- FORTRAN 77
- FORTRAN 90
- QBASIC
- Visual BASIC
- Turbo PASCAL

Code Generation

You have the option of generating only the function code, or a full code option that contains simple calling code to test the function. The **Full Test Code** option writes the function, any auxiliary function such as the error function or the Chebyshev or Fourier series evaluators, and simple calling code to test the equation. The **Function Code Only** omits the code that tests the function.

You must specify the name of the function, subroutine, or procedure for the curve-fit equation. After the file is written, you are presented with a **Generated Code Display**. The generated code will normally contain 15-16 digits precision and will use double precision (64-bit) variables. The C (long double) and PASCAL options will use long double precision (80-bit) and will be written to 19 digits precision.

- The Visual Basic full test code is specific to Microsoft Visual Basic. The TableCurve 2D Visual Basic code generation produces a Visual Basic Form [.FRM] file.

User-Defined Functions

While code is not generated for user-defined functions, TableCurve 2D will write the current UDF into the generated code as it appears in the program. This should assist you in the manual coding that is required. If your UDF uses a special function such as a Bessel or one of the statistical functions, you will have to supply your own version of the function.

No Constraint Code

Note that this generated code will contain only the native mathematical computation. There will be no conditional statements to manage undefined regions. For non-linear equations, however, constraints will appear as comments.

Adding a Visual Basic Module to an Excel Sheet

If you are using a later version of Excel that supports VBA (Visual Basic for Applications), you can **add a BASIC module** containing the curve-fit equation to the Excel spreadsheet.

Covariance Matrix Export

The covariance matrix for the current fit can be saved to a file for use in other programs. The [Save Covariance Matrix](#) option in the File menu of the [Curve-Fit](#) graph saves the covariance matrix as an ASCII file and then displays the data in this file using TableCurve 2D's [text file viewer](#).

The covariance matrix is the inverse matrix multiplied by the square of the standard error of fit (MSE). The diagonal elements thus contain the variances for the individual parameters. The off-diagonal elements contain the covariances between the parameters.

If you wish to import the covariance matrix into a spreadsheet, simply select the [Copy](#) option in the Edit menu of the [text file window](#) that displays the file contents. You may need to use a Paste Special or similar option within your spreadsheet and then specify the WK1 clipboard format for the import.

Data Summary

The [Data Summary](#) is used to display a point-by-point data summary for the current equation with the X and Y value of each data point, the predicted value, the residual and %residual, the confidence limits, and the prediction limits.

The Data Summary is toggled on and off by the [Data](#) button in the [Curve-Fit Graph](#) window. The window size and position you choose for the Data Summary is automatically saved across sessions. The Data Summary uses the [TableCurve Text Viewer](#) to display, modify, and print the information.

Options Menu

The [Options](#) menu is used to select the display of:

- [Residuals](#)
- [Confidence Limits](#)
- [Prediction Limits](#)
- [All Information \(Residuals, Confidence Limits, Prediction Limits\)](#)

You must select a small font size or landscape mode in order to fully print all three types of information on a standard size page.

The initial confidence level will be the value set in the [Confidence Intervals](#) option of the [Curve-Fit graph](#). To change to a different confidence level, select from one of the five confidence level items in the [Options](#) menu:

- [90%](#)
- [95%](#)
- [99%](#)
- [99.9%](#)
- [99.99%](#)

Window Type

If RTF windows are set in the [Preferences](#), the Data Summary uses [RTF windows](#) if there are 128 or fewer data points in the table. For larger table sizes, the Data Summary uses the much faster [standard text windows](#).

Equation List

The [Equation List](#) window displays a selection list of the curve-fit equations. This list will initially be sorted by the criteria set in the [Curve-Fit Preferences](#) Process menu option. This list can subsequently be sorted by any one of TableCurve 2D's goodness-of-fit sort criteria and its content can be filtered to contain only certain types of equations.

By default, rational equations detected as having a pole in the range of the X-data are initially excluded.

The Equation List window has its own menu. You can also access its content-related functions via the [List](#) menu in the [Curve-Fit Graph](#) window. The Equation List is toggled on and off by the [Eqn. List](#) button in the [Curve-Fit Graph](#) window.

The window size and position you choose for the Equation List are automatically saved across sessions. TableCurve 2D saves two Equation List window positions and sizes, one for the [Graph Start](#) option and one for the [List Start](#) option. Options to save the Equation List to a file, to display the List Window in color, and to print all or part of the list are present only in the List Window's own menu and not in the List menu of the Curve-Fit Graph. To access these functions when the Equation List is too small to have a menu, maximize the Equation List window.

Rank

The first item in each line of the equation list will be the ranking of the equation by the goodness-of-fit criteria currently set. Initially, all successfully fitted equations are shown except rational equations which have an undefined region in the X-range of the data. These rationals can be included in the list by toggling the [Rationals Fully Defined Xmin-Xmax](#) menu item. The rank will always reflect the position within all fitted equations, regardless of how the list is filtered.

Goodness-of-Fit

The second item will be the goodness-of-fit value. This may be the r^2 coefficient of determination, the DOF-Adjusted r^2 , the Fit Standard Error, the Maximum Error, or the F-statistic. The current goodness-of-fit criteria will be shown in the window's title bar. When the [All Goodness-of-Fit in List](#) option in the List menu is checked, all four goodness-of-fit statistics will be displayed within the equation list.

Floating Point Demand

The third item will be the FP or floating point demand for the TableCurve 2D generated language code for this equation. A value of 1 corresponds to 1 floating point addition. The higher the number, the greater the execution time for the equation code.

Equation Number and Equation

The last items in an equation list entry will be the TableCurve 2D equation number and the equation expression. The format for non-linear equations can be set in the [preferences](#).

Selecting an Equation

If you make a selection from the Equation List using a mouse, an equation is selected as soon as you click on it. All Review windows currently open will be updated to reflect the new equation. If the Equation List is the currently active window, you can also move through the list using the cursor keys and make a selection by pressing Enter. If you maximize the Equation List window, regardless of Review startup option, the Equation List window is automatically closed when a selection is made.

File Menu

The [Save As](#) option is used to save the Equation List to a simple ASCII file, to a Lotus WK1 file, to a Lotus WK3 file, to a fully formatted Rich Text Format (RTF) file, or to a partially formatted Wordstar file which retains superscripts and subscripts. In the case of the ASCII text, WK1, and WK3 files, superscripts are not preserved. A save to the RTF format seeks to exactly preserve the appearance of the equation list when imported into word processing and desktop publishing products. This option preserves columnar formatting, font colors, bolding, superscripts, and subscripts. In order to save a black and white RTF text file, uncheck the [Color](#) option in the List menu.

Use the [Printer Setup](#) item to select the current printer and optionally to configure the printer's own driver.

The [Print](#) item is used to print either the currently displayed contents of the list or the entire list. The [Text Print](#) dialog is used to customize the printing of the information.

Edit Menu

The [Copy](#) option copies the full equation list, memory permitting, to the Windows clipboard as unformatted ASCII text, as formatted Rich Text Format (RTF) text, and in the Lotus WK1 format. The unformatted text format is space delimited and is used when you to paste the equation list into a text-based program. The RTF format seeks to exactly preserve the appearance of the list when pasted into word processing and desktop publishing products. The RTF format preserves columnar formatting, font colors, bolding, superscripts, and subscripts. In order to save a black and white RTF text file, uncheck the [Color](#) option in the List menu prior to the Copy. For some products where RTF is the default clipboard format, such as Microsoft Word, use a Paste As, Paste Special, or similar option in order to specifically select the unformatted text. When you paste the equation list into a spreadsheet, the WK1 format is used.

The [ASCII Editor](#) option copies as much of the equation list as possible into the TableCurve 2D ASCII editor. On systems with 16-bit edit controls, the maximum size is about 700-800 equations. In all of the Edit menu options, the superscripts in the equation strings will not be preserved.

List Menu

If the [Keep List After Selection](#) is checked and the Equation List window is not maximized, the Equation list will not be removed after a selection is made. If this [Keep List After Selection](#) is not checked, or if the Window is maximized, the Equation List window is closed after each selection.

When the [All Goodness-of-Fit in List](#) option is checked, all four goodness-of-fit statistics will be displayed within the equation list. First will be the r^2 coefficient of determination, then the DOF-Adjusted r^2 , the Fit Standard Error, and finally the F-statistic.

The style items include setting a font for the list. This [Font for List](#) font selection is unique to the Equation List. The [Color](#) option toggles the display of the list between color and black and white.

Filter Menu

All of the Filter Menu items are also in the [Curve-Fit Graph's](#) List menu. You may apply eight different filter options to the equation list so that it contains [All Equations](#), only the [Simple Equations](#), only [Polynomial/Rational Equations](#), only [Polynomial Equations](#), only [Rational Equations](#), only [Non-Linear Equations](#), only [Continuous Trend First Derivative](#) equations (those which have a continuously increasing or decreasing first derivative across a specified X-range), or [Constant Sign First Derivative](#) equations (those which have a constant sign first derivative across a specified X-range). These options reset the selected equation to the first within the list.

For the two derivative scanning options, you can use the [Set X Range for Derivative Filtering](#) option to

specify the X range as well as how many sampling points are to be used. The default scan will range from Xmin to Xmax+1/2 Xrange with 32 points. The larger the number of sampling points, the likelihood of an equation with an undefined region or pole slipping into the list will be lessened. The procedure is a linear one, meaning that it will take twice as long to scan the list when 64 sampling points are used.

By default, rational functions are not displayed if they have a pole or undefined region in the range of the X data. The [Rational Defined Xmin to Xmax](#) option is used to toggle this additional filter.

Sort Menu

All of the Sort Menu items are also in the [Curve-Fit Graph's](#) List menu. You can sort the current list at anytime [by \$r^2\$](#) coefficient of determination, [by DOF-Adjusted \$r^2\$](#) , [by Standard Error](#) (Root MSE), [by Maximum Error](#), [by F-statistic](#), [by Floating Point Speed](#), or [by Equation Number](#). If you want the best possible least squares fit with regard to the number of coefficients, use either the DOF-Adjusted r^2 or the Standard Error criteria. If you want the equation that represents the best possible model for describing the data, use the F-statistic. The F-statistic sort will often bring many simpler equations nearer the top of the list if the equations do a respectable job describing the data using a modest number of coefficients. A sort resets the selected equation to the first equation in the list.

The [Arrange by Parameter Count](#) option uses the coefficient count as the primary sort key, and the selected goodness-of-fit sort criterion as the secondary key. When this option is selected, all of the two parameter fits will be at the top of the list and all of the twenty-one parameter fits will be at the bottom. Within each parameter count grouping, the equations will be sorted by the currently selected goodness-of-fit.

Equation References

The [Equation References](#) option, in the [Curve-Fit graph's](#) control panel and Reference menu, controls the addition and removal of equation references. These references overlay the current curve-fit and are usually used for graphical comparison with other curve-fits in the current equation set, or with fits saved from a prior analysis.

References

Up to four curve-fit references can be active at a given time. Each reference and its corresponding title will be displayed with a specific color that can be set in the [Customize Colors](#) option. The current curve-fit equation uses the [Component 1](#) color. The references use [Component 2](#) through [Component 5](#) colors.

When a reference is from the current set of candidate equations, the title will contain the r^2 value, equation number, and equation description. When the reference is imported from a previous fit, the title will contain the file name (minus the path), equation number, and equation description. Note that the standard titles with the goodness-of-fit statistics and parametric information are not present when one or more references is active.

References can be processed individually, or as a set.

Single Reference Options

Check the [Add Current Curve-Fit Equation to References](#) option to add the present curve-fit to the reference set. The reference is automatically added to the next available position. If four references are already specified, this option will be grayed. When this option is selected, the new reference will fully mask the current curve-fit (the current fit and reference are identical until a different equation is selected).



Note: a reference can be directly added without opening this Equation References option. Click the [Add Current Equation as Reference](#) button or select the [Add Current Equation as Reference](#) item in the Reference menu of the Curve-Fit graph.

Check the [Add One or More Imported Fits to References](#) option to import fit reference files from disk. These files, which are saved in a binary format and contain a single reference, have a default REF extension. To select multiple files in the import procedure, hold down the Shift or Control keys.

Check the [Save Current Equation as Reference](#) option to export the current equation and parameters as an REF fit reference file. Note that this saves only the current equation being reviewed. Note that you can also initiate this export by using the [Save Current Equation as Reference](#) item in the Reference menu of the Curve-Fit graph.

Check the [Remove Specific Reference](#) option and select the reference number desired in order to fully delete a reference.



Note: It is not necessary to remove references when you want to only temporarily cease displaying them. The [Toggle Display of Equation References](#) button or the [Display References](#) item in the Reference menu of the Curve-Fit graph turns the display of references on and off without altering them.

Multiple Reference Options

Check the [Import Reference Library](#) option to import an equation reference set. These files, which are also saved in a binary format, contain anywhere from one to four references and have an EQR default extension. The import will first remove existing references, if present. Note that you can also initiate this

import by using the [Import Reference Library](#) item in the Reference menu of the Curve-Fit graph.

Check the [Save Current Set of References](#) as Library option to export the current set of references to disk. Note that once these equation references are saved to disk and then imported, all links to the current data set are lost. This means that the titles will no longer contain goodness-of-fit statistics. Note that you can also initiate this save by using the [Save Current Set of References as Library](#) item in the Reference menu of the Curve-Fit graph.

Check the [Remove All References](#) option to permanently delete all installed references.

Creating Standards

To make a standard, you may want to combine multiple data sets, fit them simultaneously to the target model, and then save the selected fit to disk as an equation reference. When fitting future sets, the standard can readily be imported for comparison.

Evaluation Procedure

The [Evaluation](#) procedure offers extensive numerical evaluation of parametric and non-parametric models. Use this option to find function evaluations, roots, derivatives and cumulative areas. This option's Generate feature enables the creation of any data table based upon the equation being evaluated.

Single Data Input

This aspect of the Evaluation requires individual data input from the keyboard. Enter a value in the X, Y, or X2 field and then click on the desired function operation.

The following operations are available:

$y=f(x)$ Evaluate the function at X.

$x=root(y)$ Find all roots (X-values) at Y.

$y=d_1(x)$ Evaluate the first derivative at X.

$y=d_2(x)$ Evaluate the second derivative at X.

$y=\int^x$ Evaluate the area from the lower bound of the data range (or function plot range) to X.

$y=\int_x^{x_2}$ Evaluate the area from X to X2.

The result is placed in the Evaluation Table. This table is a sorted list window that accumulates all manually or automatically generated data. The table lists the data element, the X value, the Y value, and the type of function operation that produced the data.

The list can be sorted by X value, Y-value, evaluation type, and by magnitude of the confidence interval. Click on the desired table header to initiate a sort. A second click on the same heading reverses the order of the sort.

Precision

The number of significant figures used in the table can be set anywhere from 3 to 15 using the [Precision](#) option.

Confidence or Prediction Limits

If you are evaluating a Curve-Fit equation, you can also check the [ConfLim](#) or [PredLim](#) items. In these instances, the confidence or prediction limits will be computed for the curve at the X,Y listed and these will be included in two additional columns. The initial [confidence level](#) will be that interval limit currently set in the Review. The [Level](#) drop down box allows any of the confidence levels to be set.

Areas

The cumulative area computations use a Gaussian Quadrature procedure. The precision attained is shown in brackets. A minimum fractional error of 1E-8 is sought.

Generate Table



The [Automatically Generate Table Entries](#) option is used to automatically create either the evaluation table or a file from the function being evaluated. You have the option of generating the input values by specifying the starting value, increment (or count), and ending values, or you can read the input values as a column from any ASCII, Lotus, Excel, SigmaPlot, Quattro Pro, dBase, DIF, SPSS, or Systat file. The computed values can be $Y=F(X)$, $X=\text{Root}(Y)$, $Y=dF(X)/dY$, $Y=d^2F(X)/dY^2$, or $Y=\text{Area}(X_{\min}, X)$. You can use this option to generate a file or to fill the evaluation table, up to a limit of 16384 entries.

Note that a Windows sorted list is quite slow when a large number of entries are being generated. In such an instance, it may be advantageous to generate the data directly to file.

Clearing the Evaluation Table



To clear the contents of the Evaluation table, use the [Clear All Table Entries](#) button in the button panel.

Saving the Evaluation Table Data



To save only the X and Y columns of the Evaluation table, use the [Save Data to Disk](#) button. You can save the data to an ASCII file, to WK1, WK3, Excel 95, or Excel 97/2000 spreadsheet files, to an SPSS SAV file, or to a Systat SYS file. The ASCII files are written to 17 digits of precision. The binary formats offer 15 digit precision except for the WK3, which offers 19 digits. These files can be subsequently imported into TableCurve 2D's data table.

Copying the Evaluation Table Data to the Clipboard



To copy only the X and Y columns of the Evaluation table to the Windows clipboard, use the [Copy Data to Clipboard](#) button. This copies the X-Y data as 17 digit precision space-delimited ASCII text, in a 15 digit precision WK1 format, and in a full 19 digit precision WK3 format.

Listing/Printing the Evaluation Table



The [List Data](#) option places the evaluation table in a [TableCurve 2D text window](#) that can be printed and edited.

Saving an Evaluation Sequence



An evaluation sequence can be saved to disk for re-application to any new model or equation. Use the [Save Evaluation Sequence to Disk](#) button to save an EVS (evaluation sequence file). These are binary files that can only be created from within TableCurve 2D.

When the Evaluation is closed, you will be asked if you want to save an unsaved evaluation sequence only if manual entries have been made. This query is not made for automatically generated tables.

Importing an Evaluation Sequence



An EVS file can be imported using the [Import Evaluation Sequence from Disk](#) button. When a sequence is imported, the existing entries in the table are cleared. If manual entries have been made, you will be presented an option to first save the existing evaluation sequence.

MS Word/RTF Export



The **MS Word/RTF File Export** option is used to save the current evaluation table to either a MS Word file or a portable RTF (Rich Text Format) file.

Excel Export

You can export TableCurve's [Curve-Fit](#) to an Excel XLS spreadsheet file. Use the [MS Excel Export](#) button in the control panel of the [Curve-Fit graph](#) or the [Save Excel](#) option in the File menu of the [Curve-Fit graph](#). Three formats are offered varying from all curve-fit information to simple generated XY data.

XLS Format

Three Excel formats are supported:

- [Excel 2000/Excel 97](#)
- [Excel 95](#)
- [Excel v4](#)

The [Excel v4](#) format uses fixed columns, regardless of whether or not data are present. The newer Excel formats contain columns only when that type of data is present. The newer Excel formats also use a different column order and structure to be compatible with the Excel export offered in TableCurve 2D's [automation](#) facility.

Structure of Exported XLS

The [Full Worksheet](#) item writes all possible information to the worksheet including fit parameters, fit statistics, the original data with predicted values and confidence and prediction limits, and a generated section which contain X values determined by the values in the [X Start](#), [X Incr](#), and [X End](#) fields. This generated section contains the predicted Y value as well as confidence and prediction intervals. The initial values in the X generation will produce a block of 100 points. The confidence value used will be the value currently set in the [Intervals](#) option.

The [Generated Information](#) option writes only the generated columns, including the confidence and prediction limits.

The [Generated XY Only](#) option writes only the X,Y information defining the curve to the first two spreadsheet columns.

Equation Formula

The XLS worksheet produced in the [Full Worksheet](#) option can contain a generated Excel formula representation for the curve-fit equation. To enable absolute A1 cell addressing in a variable column export, the Excel 95 and Excel 2000/Excel 97 options place this formula in the G and H columns of the worksheet. For the Excel v4 format, the formula will be in the last column written in the sheet.

You must enter this cell for Excel to process the formula and you must copy this formula to all cells in the column for which you wish to compute a predicted Y value. The X values for this formula are entered in the preceding column. Note that this generated equation will contain only the native mathematical formula. There will be no conditional statements to manage undefined regions.

If [Omit](#) is checked, no formula is written. If [Write Using A1 Cell Addressing](#) is checked, the formula is written for specific cell addresses using the A1 format. If [Write Using R1C1 Cell Addressing](#) is checked, the formula is written for relative row-column addresses using the R1C1 format. The relative addressing is used in the Excel export produced by the [automation](#) facility when multiple export streams appear on each sheet. For the Excel v4 format, the options to omit the formula and select cell addressing will be grayed.

Note that no equation formula will be generated for Chebyshev and Fourier Series equations because it is

not possible to generate a single-line formula for these models. Also, for Excel 95 and Excel v4 formats, no equation formula will be generated for equations containing greater than 11 coefficients. This is due to a formula length limit built into these versions of Excel.

Inserting BASIC Module in Excel

For later Excel versions supporting VBA (Visual Basic for Applications), you may want to add a BASIC module containing the curve-fit equation to the spreadsheet. Although it involves several steps, this approach has the advantage of producing a viable formula for all TableCurve 2D equations, including the Chebyshev and Fourier Series equations. For Excel 95 and prior, it also manages equations containing greater than 11 coefficients.

This approach makes the curve-fit function global to Excel just as if it were a built-in Excel function. Further, the BASIC code is generally easier to understand than the single line formula since the coefficients appear directly rather than as cell references.

Note that you can use this option even if you have no experience with BASIC because no language understanding is needed.

The necessary steps are as follows:

In TableCurve 2D:

- **Generate Function Code** using Visual Basic or QBASIC, [Function Code Only](#).
- Generate the Excel file using this Excel export option

In Excel:

- Open the generated Excel file
- [Tools/Macro/Visual Basic Editor](#) to open the VB editor
- [File/Import File](#) with the generated BASIC file name to add the VB module to the worksheet
- Use the generated function name in formulas anywhere within the Excel sheets as in =eqn8010(A1)

Goodness-of-Fit Statistics

TableCurve 2D uses four common goodness of fit statistics. In the following descriptions, **SSM** is the sum of squares about the mean, **SSE** is the sum of squared errors (residuals), **n** is the total number of data values, and **m** is the number of coefficients in the model. **DOF**, the degree of freedom, is n-m.

Coefficient of Determination (r-squared)

$$r^2 = 1 - SSE/SSM$$

Degree of Freedom Adjusted Coefficient of Determination

$$DOF\ r^2 = (1 - SSE*(n-1))/(SSM*(DOF-1))$$

Fit Standard Error (Root MSE)

$$StdErr = \sqrt{SSE/DOF}$$

F-statistic

$$F-stat = ((SSM-SSE)/(m-1))/(SSE/DOF)$$

As a fit becomes more ideal, the r^2 values approach 1.0 (0 represent a complete lack of fit), the standard error decreases toward zero, and the F-statistic goes toward infinity.

Harvard Graphics Export

You may export TableCurve's **Curve-Fit** to a Harvard Graphics chart file. The file is created via the **Save Harvard Graphics** option in the File menu of the Curve-Fit Graph. The generated [CHT] will contain a curve-fit graph containing the data and curve-fit equation. TableCurve will copy current log axis settings as well as X and Y titles to the generated Harvard Graphics chart.

Lotus 123 Export

You may export TableCurve's [Curve-Fit](#) to a Lotus WK1 spreadsheet file that can be read by all versions of Lotus, Excel, Quattro, SigmaPlot as well as many other programs. The file is created via the [Save Lotus 123](#) option in the File menu of the Curve-Fit Graph.



A WK1 without graphs is also copied to the clipboard in the [Copy Data to Clipboard](#) option.

Structure of Exported WK1

This Lotus file will contain a section with fit parameters and statistics, a section with the original data with predicted values and confidence and prediction limits, and a generated section containing the X values you specify via the Starting X, Increment, and Ending X values. The default values will produce a block of 100 generated points. This generated section will contain the predicted Y value as well as confidence and prediction intervals. The confidence value used will be the value currently set in the [Intervals](#) option.

Named Graphs

You may elect to have 8 XY data named graphs and 5 Generated named graphs added to the worksheet. If you wish to have graphs automatically generated for Excel, you should use this option.

Equation Formula

The last column in the WK1 will contain a generated formula representation for the curve-fit equation. You will need to remove the quotation character in this cell and then copy this formula to all cells in the column for which you wish to compute a predicted Y value. The X values for this formula are entered in the preceding column. Note that this generated equation will contain only the native mathematical formula. There will be no conditional statements to manage undefined regions.

No equation formula will be generated for Chebyshev and Fourier Series equations since it is not possible to generate a single-line formula for these models.

Mark Excluded Points for Refit

When exploring a model that is appropriate for a given data set, outliers may be apparent. When this occurs, it is often desirable to immediately tag the outliers and refit the data excluding these points. The [Mark Excluded Points for Refit](#) option is used to automatically select outliers or points where math errors occur. The outliers can also be tagged manually.

Manually Marking Outliers

The [Mark Excluded Points for Refit](#) option is not needed to manually select outliers. To mark a point for exclusion in a refitting procedure, left click on the point desired. The point will be redrawn with a surrounding circle. To unmark a point that has been marked, click once again with the left mouse button. The surrounding circle will be removed.

Although this option is generally used to mark and remove outliers that were previously fitted, it can be used to mark inactive points that were excluded from the original fitting. In this instance, a refit will activate the previously excluded points.

Automatically Marking Outliers

The [Mark Excluded Points for Refit](#) option offers the following means for automatically marking outliers:

- None (Clear All Marked Points) - All marked points are cleared. The current active and inactive state of the points is locked.
- All Points with Math Errors
- All Points Outside 2, 3, or 4 Standard Errors
- All Points Outside 90, 95, 99, 99.9% Prediction Interval

Refitting



If one or more points is currently marked, one of the refit buttons will be active within the control panel of the [Curve-Fit graph](#). This button is adjacent the [Mark Excluded Points for Refit](#) button. When a refit is initiated, all marked points are toggled in state and a full refit is made. Marked active points are made inactive while marked inactive points are made active. It is thus possible to go back and forth between including and excluding a set of marked points by successive refits.

Note that a refit may result in the previously selected equation not being available. There may also be more or fewer equations in the list after a refit. When points are toggled off, a rational function that was previously defined fully across the X-range may have undefined regions or poles in the vicinity of the points toggled off. The equation may still be available if the [Rational Defined Xmin to Xmax](#) filter is disabled in the [Equation List](#), but the new fit may be of dubious value.

If a refit fails completely so that no equation is available, the original state of the data is restored.

A refit completely processes all equations previously fitted. This may require some time with large data sets, depending on the non-linear equations selected.

Numeric Summary

The **Numeric Summary** is used to display all numerical information associated with the curve-fit for the current equation. This summary includes **goodness of fit criteria**, coefficient standard errors and confidence limits for the fitted parameters, measured values such as area and extrema of the function and its derivatives, the fitting method used, an analysis of variance, and data table statistics.

The Numeric Summary is toggled on and off by the Numeric button in the **Curve-Fit graph** window. The window size and position you choose for the Numeric Summary is automatically saved across sessions. The Numeric Summary uses the **TableCurve Text Viewer** to display, modify, and print the information.

The Numeric Summary offers four **Options** menu items for controlling the amount of information presented:

- **Probabilities** - adds probability values to coefficient summary and analysis of variance table
- **Area and Extrema** - adds function area and measured function and derivative extrema to summary
- **Analysis of Variance** - adds ANOVA table to summary
- **Data Table Statistics** - adds basic data table statistics and titles to summary

The initial confidence level will be the value set in the **Confidence Intervals** option of the **Curve-Fit graph**. To change to a different confidence level, select from one of the five confidence level items in the **Options** menu:

- 90%
- 95%
- 99%
- 99.9%
- 99.99%

Goodness of Fit Statistics and Coefficient Summary

The standard numeric summary information includes the four **goodness of fit statistics** used through TableCurve 2D and a fitted parameter table that includes standard errors, t-values (parameter divided by std. error), and confidence limits.

Rank 14 Eqn 8051 Gamma(a,b,c,d,e)

r ²	Coef Det	DF Adj r ²	Fit Std Err	F-value	
0.9930667028		0.9926939449		2.6723443324	3365.9407391

Parm	Value	Std Error	t-value	95% Confidence Limits		P> t
a	10.21013877	0.401680699	25.41854460	9.412592322	11.00768521	0.00000
b	98.51427854	0.910244052	108.2284232	96.70696764	100.3215894	0.00000
c	3.682986825	0.012626218	291.6935934	3.657917174	3.708056477	0.00000
d	0.801518981	0.027243214	29.42086668	0.747426940	0.855611022	0.00000
e	1.806320747	0.052683021	34.28658230	1.701717373	1.910924121	0.00000

Area and Function and Derivative Extrema

The area is computed using a successive Gaussian Quadrature integration from the lowest active X value in the data set to the highest active X active. The precision attained in the area computation is also shown. One dimensional minimization is used to determine the minimum and maximum function, first derivative, and second derivative values and the X-values where these extrema occur. The derivative

values will be based upon analytic derivatives if these are set in the [Reference](#) menu of the [Curve-Fit](#) graph.

Area Xmin-Xmax	Area Precision		
287.24061570	2.995478e-18		
Function min	X-Value	Function max	X-Value
13.221606759	9.9090910000	105.51788491	4.0632864709
1st Deriv min	X-Value	1st Deriv max	X-Value
-76.43073838	4.7957195384	76.430738375	3.3308490285
2nd Deriv min	X-Value	2nd Deriv max	X-Value
-172.0467226	4.0632847674	76.777625516	5.3318999291

Rational Function Singularities

The poles of rational functions, which represent the roots of the denominator, are also reported in the numeric summary.

When the rational's denominator is a polynomial in x, all real roots are found and reported using an eigenvalue procedure. These are broken down by those within the X range of the data and by those anywhere outside of such. For example:

Singularities [Data Range]
 3.0433089735 6.7586369700 18.155326487 26.402880316
 Singularities [All Other]
 30.691233986 36.520959952

When the rational denominator is not a polynomial in x, such as is the case for $\ln(x)$, half order, even order, Chebyshev, and Fourier series rationals, a brute force partitioning is used to isolate pertinent real roots. 100 linear partitions are used within the data range, 50 more in a half-length range below the minimum X, and 50 more within a half-length range above the maximum X. For example:

Singularities [Data Range]
 6.6040821944 20.327000216 27.751943411
 Singularities [$\pm\frac{1}{2}$ Data Range]
 -6.604081091 30.074267286

As with all root-finding algorithms, this approach is not absolutely foolproof. The reported roots will most definitely exist, but there is a finite possibility that a root may go undetected. You can successively zoom in regions of concern to check for undetected poles. You can also use the Review's [Evaluation](#) procedure to generate up to 16384 function evaluations between any two X-values of concern. You should generate directly to file and then [Import](#) the file to graphically check for a pole that may have eluded detection.

ANOVA

The analysis of variance list includes the procedure, matrix method, minimization, iterations, and SVD condition number, depending on type of fit. The goodness of fit statistics not included in the ANOVA are then listed along with the maximum error (the largest residual) in the fit. The ANOVA table follows.

Procedure	Minimization	Iterations			
LevMarqdt	Least Squares	12			
r ² Coef Det	DF Adj r ²	Fit Std Err	Max Abs Err		
0.8890484160	0.8843270720	10.633870945	32.682979967		
Source	Sum of Squares	DF	Mean Square	F Statistic	P>F
Regr	86079.212	3	28693.071	253.743	0.00000
Error	10742.525	95	113.07921		
Total	96821.737	98			

Pure Error and Lack of Fit

When there are multiple Y observations at a specific X value, the variance in these Y values can be said to consist of pure or random error. When such replicates are present in a data set, it is possible to separate the portion of the variance attributable to pure error from that portion associated with the fitted model. This expanded analysis of variance will be automatically included in the numeric summary when such replicates are present and all weights remain set at unity.

Procedure	Minimization	Iterations			
LevMarqdt	Least Squares	7			
r ² Coef Det	DF Adj r ²	Fit Std Err	Max Abs Err		
0.9946085954	0.9944681942	2.0290264352	7.0584260241		
r ² Attainable					
0.9973156573					
Source	Sum of Squares	DF	Mean Square	F Statistic	P>F
Regr	146582.8	4	36645.699	8901.18	0.00000
Error	794.57102	193	4.1169483		
Total	147377.37	197			
Lack Fit	398.95966	94	4.2442517	1.06211	0.38327
Pure Err	395.61136	99	3.9960743		

The **r² Attainable** is the maximum r² that can be achieved with any model. The **Pure Err** sum of squares will reflect that portion of the sum of squared residuals attributable to pure error in these repeat observations. The **Lack Fit** sum of squares is simply the difference between the overall sum of squared residuals and this pure error sum of squares. When this lack of fit F-statistic is significant (>>1), the model may be inadequate. In such cases, the residuals should be checked closely for systematic trends which would confirm the insufficiency of the model.

Parameters Export

This option is used to write the parameters of the current **Curve-Fit** equation to an ASCII file. The file is created via the **Save Parameters** option in the File menu of the Curve-Fit graph. You have an option to append the coefficients to an existing file.

This option is used primarily to write the current equation coefficients, with maximum precision, to an ASCII file. The file can be written with identifiers, and if chosen, each line will be preceded by a string similar to a= followed by a space and then the coefficient. This type of simple ASCII format is identical to the INI format used in Windows applications. You may also choose to add date, time, and file source strings, the three title strings, the equation string, and the four goodness of fit statistics. These files are intended for import by your own programs and by the various math programs. After the file is written, you are presented with a **Parameter File** display.



Precision Summary

The [Precision Summary](#) is used to determine how much precision is preserved in the current equation when the successively fewer digits of precision are used in the coefficients. The error is measured at Xmin, Xmean, and Xmax. For linear equations, the impact of removing (zeroing) each parameter is also shown. The Precision Summary is toggled on and off by the [Precision](#) button in the [Curve-Fit Graph](#) window. The window size and position you choose for the Precision Summary is automatically saved across sessions. The Precision Summary uses the [TableCurve Text Viewer](#) to display, modify, and print the information.

The Precision Summary is recommended when contemplating the use of [Chebyshev converted coefficients](#).

Residuals Graph

The residuals can be displayed in five different formats from within a [TableCurve 2D Graph](#):



[Basic Residuals](#) - the simple difference between the Y data value and the Y predicted from the curve fit



[Percent Residuals](#) - the residuals as a % of the Y data value



[Standardized Residuals](#) - the residuals as a fraction of the fit's standard error



[Distribution](#) - the residuals in a binned histogram



[Delta SNP](#) - the residuals as a delta stabilized normal probability

These options are available in the [graph toolbar](#) and to the left side of the dialog.

By default, the [Basic Residuals](#) graph can also double as a standardized residuals graph since the [Point format](#) can specify the coloring of residuals by fit standard error.

The Residuals Graph is toggled on and off by the [Residuals](#) button in the procedure. You may also close the Residuals Graph directly. The window size and position you choose for the Residuals Graph is automatically saved across sessions.

For curve-fit residuals graphs, the runs count is displayed in brackets within the default Y-axis graph title. The runs count is the number of sign changes that occur within the residuals across the X data range. Systematic trends within residuals generally result in a lower runs count.

Residuals Distribution Graph

The least-squares coefficient standard errors and confidence ranges as well as the curve's confidence and prediction intervals reported by TableCurve 2D contain an implicit assumption that the residuals are normally distributed. These uncertainty statistics cannot be assumed correct unless this condition of normality is verified.

The [Distribution Graph](#) option displays a binned histogram of the residuals. Distributions with obvious asymmetry or wide tails would readily disqualify this assumption of Gaussian errors.

TableCurve 2D requires at least 16 active points in order to produce a residuals distribution. Note that any histogram is of dubious merit when data table sizes are small because of the large bin spacings. The greater the number of data points, the more accurate the distribution will be.

Delta Stabilized Normal Probability Plot

TableCurve 2D offers this approach as the best way to assure errors are normal. A stabilized normal probability (SNP) plot uses an arcsine transformation on both X and Y to produce a normal probability plot that uses a linear scale for both the X and Y axes. On such a plot, perfectly normal errors plot as a 45 degree line. Critical limits also have a 45 degree slope, and lay equally above and below this line.

TableCurve 2D modifies the SNP slightly and uses a delta SNP, where the X value is subtracted from the Y. This produces a horizontal $y=0$ for pure normal data, and horizontal critical limit lines.

TableCurve 2D plots 90, 95, 99, and 99.9% critical limit lines on the SNP plot. A 99% critical limit means that in only 1 out of 100 data sets should even a single point violate this limit. You may find the 99% critical limit the most useful. If even a single data point in the SNP violates this 99% limit, it is reasonable to assume that the errors fail this normality test. The following graph is from a curve fit which yielded normal errors.

By default, the 90% critical limit lines will be blue, the 95% green, the 99% yellow, and the 99.9% red.

You should inspect the SNP before attempting to use the parameter confidence statistics or confidence or prediction intervals in any way.

For more information on the SNP, you may refer to:

- John R. Michael, "The Stabilized Probability Plot", *Biometrika*, 70,1, p11-17, 1983.
- Lloyd S. Nelson, "A Stabilized Normal Probability Plotting Technique", *Journal of Quality Technology*, 21,3, 1989.

Maximum Likelihood

When the normal assumption is invalidated due to appreciable tails in the residuals distribution, equally invalidated is the assumption that least-squares is furnishing the maximum likelihood fit. In such a case, one of TableCurve 2D's robust minimizations may represent a better maximum likelihood model.

If you choose to fit a robust model, please remember that TableCurve 2D's goodness of fit statistics are all based on a least-squares common frame of reference. As such, the goodness of fit values will fail to reflect the improvement derived from switching to a robust method. Also, you should not assume simply because a distribution of errors is Gaussian that a robust procedure is of no value. Two key reasons for using a robust minimization are to deal effectively with outliers and a wide dynamic range on the y-variable.

List



The [List Data](#) option lists the residuals. The listing uses the [TableCurve 2D text viewer](#) facility.

Copy



The [Copy Data to Clipboard](#) option copies the residuals data to the clipboard. Formats include full precision binary (for spreadsheets such as Excel) and ASCII (for pasting into text editors).

Save



The [Save Data to Disk](#) option writes the residuals data to a supported file format. These formats include ASCII, Excel 97/2000, Excel 95, Lotus WK3, Lotus WK1, SPSS, or Systat.

MS Word/RTF Export



The [MS Word/RTF File Export](#) option is used to save the current residuals graph to either an MS Word file or a portable RTF (Rich Text Format) file. The graph is inserted into the file as a Windows metafile.

Analyzing Error Models



When the normal assumption is invalidated due to appreciable tails in the residuals distribution, equally invalidated is the assumption that least-squares is furnishing the [maximum likelihood](#) fit. If you wish to see if one of TableCurve 2D's non-linear [robust minimizations](#) would possibly represent a better

maximum likelihood model, select the [Error Models](#) option. You are furnished an r^2 value for the four error models. This option is available only in the [Curve-Fit Review](#).

Improving a Fit by Adding a Residuals Fit Model



The actual residuals can sometimes be successfully fitted. This secondary model can then be added to the primary model in order to further reduce the overall standard error of fit.

Fitting the Distribution Data



The residuals distribution data can be fitted by TableCurve 2D's peak functions as yet another verification of normality.

- While this approach is one further way to test normality, you are encouraged to rely primarily on the SNP plot for assuring normality.
- It is also recommended that you use this information very cautiously when there are 10 or fewer bins in the distribution histogram.
- If you choose to fit a **robust** model, please remember that TableCurve 2D's goodness of fit statistics are all based on a least-squares common frame of reference. As such, the goodness of fit values will fail to reflect the improvement derived from switching to a robust method.
- Do not assume simply because a distribution of errors is fitted well by a Gaussian that a robust procedure is of no value. Two key reasons for using a robust minimization are to deal effectively with outliers and a wide dynamic range on the y-variable. It is very possible that such conditions will not be reflected in the fit of the overall distribution.

SigmaPlot Export

You may export TableCurve's Curve-Fit to a SigmaPlot Windows graph file. The file is created via the [Export to SigmaPlot](#) button in the control panel of the [Curve-Fit graph](#) or the [Save SigmaPlot](#) item in the Curve-Fit graph's File menu. In addition to writing the SigmaPlot file, you may optionally generate a SigmaPlot Transform for the curve-fit equation.

SigmaPlot Format

The [SP2000/v5 \(All Graphs\)](#) format writes a SigmaPlot Windows file for use with SigmaPlot 2000 and SigmaPlot Windows v5. This option contains eight graph pages, four in black and white and four in color.

The remaining options write a SigmaPlot SP5 file format. The SP5 is a DOS-based SigmaPlot format that is portable to all versions of SigmaPlot. Since this format is limited to a single graph page, the following four options are available:

- The [SP5 \(Standard Curve-Fit\)](#) graph page contains a single [Curve-Fit graph](#) of the data and curve-fit equation.
- The [SP5 \(Curve-Fit,Residuals\)](#) graph page contains the [Curve-Fit graph](#) and a bar graph of the residuals.
- The [SP5 \(Statistical Graphs\)](#) graph page contains the [Curve-Fit graph](#), a [Residuals graph](#), and a graph containing [Confidence and Prediction Intervals](#) about the curve-fit equation.
- The [SP5 \(Engineering Graphs\)](#) graph page contains the [Curve-Fit graph](#), a Derivatives graph, and a Cumulative Area graph.

TableCurve will copy current log axis settings as well as X and Y titles to the generated SigmaPlot plots.

Generated Data

The SigmaPlot curve-fit worksheet contains all possible curve-fit information, including fit parameters, fit statistics, the original data with predicted values and confidence and prediction limits, and a generated section which contain X values determined by the values in the [Starting X](#), [Increment](#), and [Ending X](#) fields. This generated section will contain the predicted Y value as well as confidence and prediction intervals. The initial values in the X generation will produce a generated block of 100 points. The confidence value used will be the value currently set in the [Intervals](#) option.

Generate SigmaPlot Transform

If the [Generate SigmaPlot Transform](#) box is checked, TableCurve 2D will generate a SigmaPlot transform with the same prefix name as the generated SigmaPlot file, and an [XFM] extension. This transform can be used to compute the predicted values for the curve-fit equation for X values that are subsequently entered in SigmaPlot. The transform code assumes the initial columns for an empty worksheet and the last columns of a TableCurve-generated worksheet when there is information in the initial column. Note that this generated transform will contain only the native mathematical formula. There will be no conditional statements to manage undefined regions. Note also, that no transform will be generated for [Chebyshev](#) and [Fourier Series](#) equations.

Editing the Templates for SigmaPlot 2000/SPW5

If you would like to modify the format, styles, and general appearance of the generated SigmaPlot graphs, or delete or add graphs, this can be easily done. You must follow these steps:

From TableCurve 2D:

- Export a SigmaPlot 2000/v5 file. If you want to add graphs that use curve-fit references, be sure that references are present before the SigmaPlot file is generated.

From the Operating System

- Back up the files [tcurvee.ovl](#), [tcurvef.ovl](#), [tcurveg.ovl](#), [tcurveh.ovl](#) that exist in the TableCurve 2D v5 directory.

From SigmaPlot:

- Open this generated file. Open the data worksheet (Data 1) within this file. Click in the upper left corner of the grid (at the intersection of the row and column titles) to highlight the entire sheet. Click the [Copy](#) button. Close the data worksheet and the generated file.
- Open the [tcurvee.ovl](#) file in the TableCurve 2D v5 directory. Open the data worksheet (Data 1) within this file. Place the cursor in row1 column1 of the worksheet and click Paste. This data is used temporarily so that graph elements are visible.
- Edit the existing graphs, delete unneeded graphs, and add new graphs. You can add as many graph pages as you like. The unusual x title and y title strings in the existing graphs are tokens that are replaced with the current x and y titles at the time the file is generated. You must exactly copy these x and y title strings to all new graphs where you want the x and y titles to be automatically generated.
- In the data worksheet, click in the upper left corner of the grid (at the intersection of the row and column titles) to highlight the entire sheet. Press the [Delete](#) key to clear all data. Click the [Save](#) button and then close the file.
- TableCurve 2D uses four templates: [tcurvee.ovl](#), [tcurvef.ovl](#), [tcurveg.ovl](#), [tcurveh.ovl](#). The first is used for linear x, linear y axes states; the second for log x, linear y; the third for linear x, log y; and the last for log x, log y states. If you want your revisions generated for the four different axes states within TableCurve 2D, you must edit all four files. If you make extensive changes to [tcurvee.ovl](#), it will be easier to import it, revise the log states of the various graphs, and then save out [tcurvef.ovl](#), [tcurveg.ovl](#), and [tcurveh.ovl](#). Be sure the saved names match and do not have JNB extensions.

UDF Generation

The [Save TableCurve UDF](#) option in the File menu of the [Curve-Fit Graph](#) is used to generate UDFs for built-in equations with 10 or fewer parameters. Polynomial and rational equations with 11 or more parameters are excluded. Also excluded are the Chebyshev and Fourier Series equations.

The generated UDF will consist of the full mathematical equation expression rather than a reference to a built-in function. Note that this generated code will contain only the native mathematical computation. There will be no conditional statements to manage undefined regions. The estimates in the generated UDFs will consist of the values of the actual fitted parameters.

Note that for non-linear equations, generated UDFs will not execute as fast as the built-in functions available via the [Non-Lin function insert help](#) in the [User Functions](#) entry dialog. The built-in non-linear functions also contain the conditional code insuring that the functions will return a valid value for all X.

Help Menu

The [Help Menu](#) offers standard Windows Help as well as a special equation help which lists all equations in TableCurve 2D's equation set.

Help Items

- Help Topics
- Linear Equations
- Non-Linear Equations
- List Equations
- What's New

Testing Automation Code

- Test Automation DLL

Author, Copyright

- About TableCurve 2D
- Copyright

Help Items

The [Help Topics](#) item in the Help menu opens the main TableCurve 2D help system. Select the [Contents](#) tab to review the contents of the help system in a book-type format. Select the [Index](#) tab to search the help system for topics that contain a given keyword.

The [Linear Equations](#) item opens the help system and displays the [Linear Equations](#) topic.

The [Non-Linear Equations](#) item opens the help system and displays the [Non-Linear Equations](#) topic.

The [What's New](#) item opens the help system and displays the [What's New](#) topic. This will mainly be of interest to v4 users.

The TableCurve 2D main help system can also be accessed by pressing F1 from the main menu or by the Help button located in most of TableCurve 2D's dialogs.

List Equations

The [List Equations](#) item in the Help menu displays all of TableCurve 2D's equations. This help can also be accessed by pressing F2 from the main menu. This help offers the means to save the equation list to a file, to change the font and to toggle color in the display, to select and optionally configure a printer, and to print either the equations currently displayed in the window or all equations.

File Menu

The [Stream to MS Word](#) option will stream the list of equations directly to Microsoft Word.

The [Save As](#) option is used to save the Equation List to a simple ASCII file, to a WK1 or WK3 spreadsheet, or to a formatted RTF file. The Exit item is used to close the Equation Help Window.

Use the [Printer Setup](#) option to select the current printer and optionally to configure the printer's driver. The [Print](#) item is used to print either the equations currently displayed in the window, or all equations. The [Text Print](#) dialog is used to customize the printing of the information.

Style Menu

Use the Style menu to select the text font and size for this specific help window or to toggle predefined colors on and off.

Tutorials

The help menu offers access to eight tutorials:

- Fitting Approximating Functions
- Fitting Parametric Functions
- Non-Parametric Interpolation
- Noise Reduction
- Derivative Estimation
- Eigendecomposition Filtering
- Fourier Filtering
- Prediction and Forecasting

These tutorials are also available in a PDF (Adobe Acrobat) format. All eight tutorials are in [tutorials.pdf](#) in the PDF subdirectory of the TableCurve 2D v5 directory. The PDF subdirectory can be deleted if PDF files are not needed.

SPSS Sales

Distributors Worldwide

SPSS Science has distributors worldwide. More information is available on the web at:

<http://www.spssscience.com>

North, Central and South America

SPSS Science Worldwide Headquarters

SPSS Inc.

233 S. Wacker Drive, 11th floor,

Chicago, IL 60606-6307

Phone: **1 (800) 543-2185**

Fax: 1 (800) 841-0064

e-mail: sales@spss.com

Continental Europe, Asia, Australia, Africa

SPSS Science Software GmbH

Schimmelbuschstr 25

D - 40699 Erkrath

Germany

Phone: **+49.2104.9540**

Fax: +49.2104.95410

e-mail: euroscience@spss.com

United Kingdom

SPSS Science Software UK Ltd.

The Research Park

Vincent Dr.

Edgbaston, Birmingham B152SQ

England

Phone **+44 (0)212-471-4199**

Fax: +44 (0)121-471-5169

SPSS Technical Support

The services of SPSS Technical Support are available to registered customers. Customers may call Technical Support for assistance in using SPSS products or for installation help for one of the supported hardware environments.

To reach Technical Support, see the SPSS technical support page on the web at:
<http://www.spssscience.com/tech>

Or contact us:

In the U.S.:

Telephone: **(510) 412-2900** (8:00 AM to 5:00 PM Pacific Time)
FAX: (510) 412-2909
Email: scisupport@spss.com
Mail: 501 Canal Blvd. Suite F, Richmond, CA 94804

In Europe:

Telephone: **49 2104 / 95480**
FAX: 49 2104 / 95410
Email: ascsupport@spss.com
Mail: Schimmelbuschstrasse 25, 40699 Erkrath, Germany

Update Product License

TableCurve 2D requires an SPSS Inc. license code in order to run. This license code is a 38-digit code furnished with the product order. The product must be enabled or re-enabled using the [Update Product License](#) option in the Help menu immediately after installation, after a license expires, or when a network or site license is updated. This option is automatically started when the product is first run and when the license validation fails for any reason.

To initiate a license for the first time, you must enter your name, company, TableCurve 2D serial number, and the 38-digit license code. The SPSS Inc. license code can be entered with or without the spaces. After the information is entered, use the [Update License](#) button to implement the new license.

To renew or revise an existing license, the previously entered name, company, and TableCurve 2D serial number will be shown. These can be optionally revised if needed.

If you are updating an existing [network](#) version, you must be certain to re-enable write permission for the installation directory shown in the dialog if you chose to make this directory read-only after the prior network installation.

Network Support

Network and Site Licenses

Multi-user network packs and site licenses enable multiple copies to be run concurrently from a single product installation. Single user copies are limited to the installed machine. The number of current and maximum allowable concurrent users is shown in the Help menu's [About TableCurve 2D](#) option. To upgrade to a multi-user network pack, please contact [SPSS sales](#).

Multiple Instances

TableCurve 2D supports multiple instances on a given machine. The first instance of the program will be the primary one, and only it will save a configuration. Additional instances are supplemental, and intended to complement the primary instance. When the first or primary instance of TableCurve 2D is closed, all other instances of TableCurve 2D running on the same machine will also be closed. In network versions, supplemental instances do not affect the user count.

Configuration Files

For network installations, the TableCurve 2D configuration files are written to the local Windows directory. Single user installations write these configuration files to the main TableCurve 2D directory. The file TC2DV5.CFG will contain all user preferences while the file TC2D2V5.CFG will contain a UDF library of the last session UDFs.

Register Product

TableCurve 2D is registered at the following Internet address:

<http://www.spssscience.com/tc2dreg.htm>

The [Register Product](#) item in the Help menu automatically accesses this web address. Follow the instructions on the web for registering TableCurve 2D v5. In order for technical support to be available and for special update pricing on future TableCurve 2D releases, it is necessary that you register the product.

About TableCurve 2D

This dialog will list:

- Current TableCurve 2D Version
- Copyright
- Your Name
- Your Company
- Your Serial Number
- Your User Position

The user position is which user you are of the maximum number of **concurrent users** supported by your copy of the product. If this is an additional instance of TableCurve 2D, you will be shown that this is a **supplemental instance**.

Function Insert Feature



A special help exists for accessing the various functions used within user entered expressions and equations. This feature is found within the [Calculation](#), [View Function\(X\)](#), [Generate Data](#), and [User-Defined Function \(UDF\)](#) procedures.

Click on the type of function you are seeking and then on the specific function of interest. The function is automatically inserted into the function string at the current cursor position. You may need to modify the symbols used as arguments in the function to match the variable intended for the expression.

The [General](#), [Trigonometric](#), [Statistical](#), [Bessel](#), [Non-Linear](#), and [PeakFit](#) functions will always be available. The [XY Data Table](#) constants will be available for calculations and UDFs. The [Calculus](#) functions which include derivatives, sums, and integrals are available in [UDFs](#) and in the [Generate Data](#) option.

The conditional IF terms are in the [General](#) group.

The [Non-Linear](#) group contains zero-intercept versions of TableCurve 2D's non-linear functions.

The [PeakFit](#) functions are a set of special peak, transition, and waveform functions used in the PeakFit product. These are made available in TableCurve 2D for UDF cross-compatibility.

TableCurve 2D Graph

All graphs within TableCurve 2D use a common interface. Each graph contains its own tool bar and status bar. All graph options are accessed from the tool bar.



Prints the current graph to an output device.



Copies the current graph or its numeric information to the clipboard.



When two graphs are present, steps through available layouts.



Sets **XY-Sectioning** mode. This mode is used to toggle XY data regions and individual points on and off.



Sets **X-Sectioning** mode. This mode is used to toggle X data regions and individual points on and off.



Sets **Process Points** mode. This mode restricts the mouse to data point positions.



Sets **Zoom** Mode. This button does not usually appear since zoom-in is always available for most graphs. When it is not, this mode must be selected in order to zoom-in with the mouse.



Sets **Point Selection** mode.



Toggles **XY Hints** on and off.



Toggles **Status Bar** on and off.



Sets fixed size **Graph Layout**.



Sets floating size **Graph Layout**.



Sets maximized size **Graph Layout**.



Splits Y and Y2 plots.



Inverts Y and Y2 plots.



Hides Y2 plot.



Hides Y plot.



Opens **Graph Scaling** dialog.



Resets **Default Scaling**.

-  Sets **Common Y,Y2** autoscaling.
-  Sets **Log X** axis.
-  Sets **Log Y** axis.
-  Sets **Log Y2** axis.
-  Includes **Inactive Points** in autoscaling.
-  Sets **2D View** (layout, split, states of titles, labels, and grids).
-  Sets **Point** formats.
-  Sets **Custom Titles** and title and label font sizes.
-  **Font selection** for current graph. Graphs are limited to a single font.
-  Select **Color Scheme** and **Customize Colors**.
-  Sets **Resolution** for Rendering Functions.
-  Opens **Confidence Intervals** dialog.
-  Toggles **Confidence/Prediction Intervals**.
-  Opens **Error Bars** dialog.
-  Toggles **Error Bars**.
-  Display actual **Residuals**.
-  Display **Residuals as % of Y**.
-  Display **Residuals as fraction of SE** (standardized).
-  Display **Residuals Distribution** (histogram).
-  Display **Delta SNP** (stabilized normal probability) plot.
-  Toggles reference data on and off.
-  Toggles reference labels on and off.

Most TableCurve 2D graphs allow a standard zoom-in using the left mouse button and a clearing of the zoom-in scaling by right clicking the mouse anywhere within the graph region except on a data point.

Confidence Intervals

Confidence and Prediction Intervals

The [Review Curve-Fit Graph](#) offers the option of displaying confidence and/or prediction bands about the fitted curve.



The [Set Confidence/Prediction Intervals, % Confidence](#) button is used to set confidence interval type and percentage.

The option opens a simple dialog in which you can individually check [Confidence Intervals](#) and [Prediction Intervals](#), and select from [90%](#), [95%](#), [99%](#), [99.9%](#), and [99.99%](#) Confidence.



Since confidence and prediction intervals assume the normality of residuals, you are strongly encouraged to use a [Residuals Stabilized Normal Probability Plot](#) to verify the validity of intervals. While important at all confidence levels, this confirmation of normality is especially critical for the higher confidence levels.



The [Show Confidence/Prediction Intervals](#) button toggles the currently set intervals on and off.

The selected confidence level applies not only to the confidence and prediction intervals in the graph, but also to the initial confidence limits used for the parameters in the [Numeric Summary](#) and to the initial confidence and prediction ranges for the data in the [Data Summary](#).

Confidence Intervals

Confidence intervals are useful for replicated data in which, for each X, there are either multiple Y observations or a single Y value which represents an average from multiple Y observations (if each Y value in the data set consists of an average from multiple observations at the same X, the point should be weighted with the inverse square of the standard deviation). With many Y observations at a given X, the average Y can be said to approach the true Y value, since the errors will sum to zero.

A 95% confidence interval is the Y-range for a given X that has a 95% probability for containing the true Y value.

Prediction Intervals

Prediction intervals are useful for predicting, for a given X, the Y value of the next experiment. It is often used when a fit represents a single experiment, where each Y value is a single observation rather than an average. In this case, the weight for each Y value isn't based upon a standard deviation from multiple observations, but rather is inversely related to the experimental uncertainty for the individual measurement, if such is known. If the uncertainty of the Y measurement is unknown or thought to be equal for all X, all points can use equal 1.0 weights.

A 95% prediction interval is the Y range for a given X where there is a 95% probability that the next experiment's Y value will occur, based upon the fit of the present experiment's data.

Local Measure of Error

Confidence and prediction intervals measure the confidence only at a specific X, not for the entire X data region. These must be computed for each X value in the curve. For fits containing models that are computationally intensive, it may take TableCurve 2D some time to compute these intervals. This is especially true with the [high precision rational and polynomial](#) models, since the intervals must also be

computed using high precision math.

Relation to Standard Error

There is often a strong similarity between points which lie outside 2 standard errors and the 95% prediction interval. These are not identical, however.

The standard error of fit is based upon the overall fit. It does not matter if a point is near a portion of the curve that is well characterized by the fit, or a point that is poorly characterized by the fit. A certain standard error in Y exists for the overall fit, and a point with a residual whose magnitude lies beyond a given multiple of this value is drawn in a certain color.

The prediction interval, on the other hand, is a more localized measure of error. Points near a region of the curve strongly determined by the fit will have a slightly tighter confidence interval than those points near a region of the curve that is only weakly determined in the fit. Generally, strongly determined regions of the fitted curve will have a good number of accurate data points whereas poorly determined regions of the curve will have relatively few points and these will have higher errors.

Drawing Intervals

If the standard uncertainties are negligible, these intervals will not be drawn. On very good fits, it may still appear that no intervals were drawn simply because they are graphically covered by the curve-fit line.

Errors in Intervals

If a math error occurs in constructing these intervals, a vertical error line of a different color from that of a standard math error will be drawn at the X-value where the error occurred. The calculation of the intervals is sensitive to floating point precision limits. When both the equation error and intervals errors are drawn, the equation itself is undefined. When only the intervals errors are drawn, the failure may be due to the limitations of machine precision. The latter case is especially true in regions outside the actual data area.

Error Bars

The option of error bars is available whenever replicate information is available for one or more data points. In TableCurve 2D, this can occur in one of two ways:

- In any **Import** option which offers **Column Selection**, a **weights** column is imported with the **SD Wts** box checked. Since the n for each entry is not imported, error bars associated with replicates are limited to multiples of the standard deviation and to intervals based on a normal distribution. Student-t distribution based confidence and prediction intervals will not be available.
- The **Process Replicates** option in the **Data menu** is used on a data table containing replicates. This option automatically searches a composite data table for replicate X values, and replaces the various instances with a single table entry where the Y value consists of the mean of the individual Y values, and the weight value consists of 1/variance ($1/SD^2$). This option also stores the n for each processed replicate entry, enabling error bars to be expressed as a Student-t distribution based confidence or prediction interval.



This graph option opens the **Error Bars** dialog.



This graph option toggles the error bars on and off.

Student-t Based Confidence Interval

The **As Conf Interval** selections offer an interval for the mean of the n entries comprising a given data point. Since the mean converges to the true value as n goes to infinity, confidence intervals are often thought of as the limits about the true value.

Student-t Based Prediction Interval

The **As Pred Interval** selections offer an interval for the next observation based on the n entries comprising a given data point.

Simple Multiples of Standard Deviation

The **As $\pm$ SD** selections offer the simple SD multiples often used for error bars.

Normal Distribution Prediction Intervals

The **As $\pm$ SD (%Pred n=Inf)** selections also offer simple SD multiples. These are based upon the prediction intervals computed using a normal distribution (a Student-t distribution with n equal to infinity).

Just as the intervals about a fitted curve require normally distributed residuals, the error bar intervals require normally distributed errors within the replicates comprising each point in order to be fully valid. Since the n for each point representing a replicate is often small, and the confirmation of normality is seldom made, error bar intervals should generally be regarded as approximate. This is particularly true for 99% and higher levels where deviations from normality are more pronounced.

Font Select

The Font item is used to set the font for the current graph or text. Only one font is available within any given graph or text window.

Font Selection

All primary typefaces in the system are shown in the selection list. When you select a font, you will see its screen representation with the attributes and size currently set. Select the bold, italic, or bold italic styles if you want these attributes applied to the font.

Color

For TableCurve 2D text windows, you can also select the color for the font.

Font Size

For graphs, this size is a default size that is used for automatic title generation. It is overridden completely by the custom sizes specified if **Custom Titles** are active. The default font size is used as the basis for the automatic titles presented. Some titles may be larger or smaller than this specified size. The actual font size displayed on screen will also be scaled in accordance with the height of the graph window.

Graph 2D View

The **2D View Options** dialog for a **TableCurve 2D Graph** itself incorporates a copy of the current graph. Changes made in any of the view options are immediately reflected in the graph. This option is used to control a graph's layout, the state of its status bar and XY hints, how the Y and Y2 axes are to be arranged, the state of all titles and labels, and the state and size of grids.

Primary Layout



The **Fixed Frame** option uses a fixed graph area that never varies relative to its position within the overall frame. This option would normally only be used for final output. Sufficient room exists to handle the font sizes of most **titles**. This option is directly available in the tool bar of most graphs.



The **Floating Frame** option seeks to create the largest possible graph area while accommodating all titles and labels. This layout seeks to preserve aesthetics, and yet at the same time, provide a large canvas for the graph information. The actual graph frame floats in size, depending on the space requirement of the various elements outside the graph frame. This option is also directly available in the tool bar of most graphs.



The **Maximum Area** option discards all external titles and labeling, using only small internal X and Y labels. This option offers the greatest possible graph area. It was designed to maximize the amount of curve-fit information. It is the default for non-output graphs in TableCurve 2D. This option is likewise available in the tool bar of most graphs.

The **Border** option controls the state of a graph's outermost bounding frame. When copying graphs to the clipboard or saving graphs as image files for importing into other applications, you may not wish to have an external border. This option is disabled on maximized area graphs since they do not have an external frame.



The **Status** option controls whether or not the graph's status bar should be displayed.



The **XY Hints** option controls whether or not the X,Y data hints appear below the cursor when moving it across graphs.

Y - Y2 Layout

These options will be available if a given graph has both a Y and a Y2 plot, and it is possible to separate the two plots.



The **Split** item controls whether the Y2 graph is to appear separately, above the Y graph, or whether the Y and Y2 plots are to be plotted within the same area.



The **Invert** item enables the Y2 graph to appear on the bottom and the Y graph to appear on top. This option is only available when the Y and Y2 plots are split.

The **50:50**, **67:33**, and **75:25** options specify the relative areas of split plots.

Titles

You may set the state for any title in a **TableCurve 2D Graph**. It is usually simpler to turn a title off than to use the **Custom Titles** option to clear it. When **All** is clicked on, all titles are toggled on. When it is clicked

off, all titles are turned off. Titles 1-5 are those above the graph. The Y and Y2 titles on both the left and right sides of the graph can be independently controlled.

Labels

You may also set the state for any axis label in a [TableCurve 2D Graph](#). Axis labels are the numeric values plotted along the X and Y axes. You may also set the label precision ([Prec](#)) from a low of 3 significant digits to a high of 8 digits.

The font size of the axes labels is set in the [Custom Titles](#) dialog.

Grids

The state of the grids used in the graph are similarly set. You may also set a grid [Level](#) from 0 to 10. A grid level of zero draws a simple line. For all other levels, the grid is drawn as a series of dots. This enhances the appearance of the grid in printed output. A level of 1 skips every other pixel. An level of 10 skips 10 pixels for each one drawn. As such, the grid intensity decreases as the level value increases.

When [printing](#), you may encounter a problem with some printers with the dot-type grids. In this instance, you will need to set a level of zero and use simple line grids.

The intermediate grids with log scaling are automatic. If a logarithmic axis has a large number of decades, however, the intermediate log grids will not be drawn.

Saving a 2D View

The [Save](#) item is used to save the current 2D View to a disk file. These are binary files with [V2D] extensions. Since each graph in TableCurve 2D contains its own settings, you may wish to save a 2D View to disk if you wish to use it for a number of the program's graphs.

Importing a 2D View

The [Read](#) item is used to update the current graph with the settings in a previously saved [V2D] custom view file.

Reset

The [Reset](#) button restores the current 2D View to the state existing when the dialog was opened.

Graph Colors

The [Select Color Scheme](#) and [Customize Colors](#) item in [TableCurve 2D Graphs](#) is used to set the colors for the current graph to one of the predefined color schemes or to select or revise a custom color scheme for the type of graph displayed. This dialog incorporates a copy of the current graph. Any color scheme selection is immediately reflected in the graph.

Note that you may print a Black and White graph from any color scheme (TableCurve 2D internally changes to the Page White BW color scheme for printing a monochrome graph). The most recent custom color scheme for each graph in the program is saved across sessions. Saved also for each graph is the most recent customized color set.

Color Selection

The various graphs within TableCurve 2D are preassigned a default color scheme. If you dislike a given color scheme, you can select one of the ten predefined schemes or build your own custom scheme.

Note that many of the color schemes are not immediately apparent if the [Graph Layout](#) is currently maximized. If you wish to modify or create a custom color scheme, you may wish to first change to one of the other graph layouts containing titles and a standard background.

Selecting or Reading a Custom Color Scheme

A custom color scheme for each graph is automatically saved across sessions. This color scheme is listed as the [User Customized](#) option. Initially, this user color scheme will contain the Page White Color scheme, the one you would most likely want to modify for color printing. To first create a custom color scheme, first select the one existing color scheme closest to the desired colors and then select the [Customize Colors](#) button. To read a custom colors [CLR] file from this initial color selection dialog, simply select the [Read](#) option.

Creating a Custom Color Scheme

The [Customize Colors](#) button opens a Custom Colors dialog. Like the color scheme dialog, it too contains a copy of the current graph which will be updated with each color modification. This dialog will display the graph colors used in the program and to the left of each you will see the current colors. Simply click on the color you wish to customize and then select one of the 16 pure Windows colors in the selection palette. The graph will immediately change to reflect your choice if that element exists within the current graph.

These 16 colors are those which are guaranteed to be present in all Windows applications, regardless of palette settings.

Saving a Custom Color Scheme

The most recent custom color scheme for a given type of graph is automatically saved across sessions. If you wish to have more than one custom color scheme for a given type of graph, such as plotter and color laser printer color schemes for peak fit graphs, you will have to [Save](#) the custom colors to a disk file and use the read item to recall them prior to printing. You will also wish to save the custom color scheme to disk if you wish to use these same colors in other program graphs. Colors files have CLR default extensions. These are binary files which can only be generated from within the program.

Reading a Custom Color Scheme

The Custom colors option also has a [Read](#) option to import custom color [CLR] files.

Reset

The [Reset](#) option restores colors to the state existing when the dialog was opened.

Graph Copy

The [Copy Graph to Clipboard](#) option in [TableCurve 2D Graphs](#) offers a variety of methods for copying the current graph or its numeric contents to the clipboard or to file.

Clipboard destination options are as follows:

- Windows Bitmap to Clipboard
- Windows Metafile to Clipboard
- Windows Enhanced Metafile to Clipboard
- All Numeric Info as Spreadsheet Block to Clipboard

File destination options include:

- Windows Device Independent Bitmap (DIB) To File
- Windows Metafile to File in Aldus Placeable Format
- Windows Enhanced Metafile to File

Controls consist of:

- Use Color
- Bounding Frame
- Preserve Superscripts, Subscripts, and Symbols in Metafiles

The [All Numeric Info as Spreadsheet Block to Clipboard](#) is the only option that does not export the graph image. This option copies all of the numeric information used to create the graph to the clipboard in a spreadsheet format. It is easily pasted into Excel, Lotus 123, Quattro Pro Windows, SigmaPlot and many other Windows applications.

All other options export either a bitmap or a metafile. Most Windows word processing, spreadsheet, and graphics software can paste in Windows Bitmaps and Windows Metafiles copied to the clipboard by TableCurve 2D.

Bitmaps

A Windows bitmap is a raster image consisting of pixel-type information. A bitmap will accurately retain the positioning of all elements within the graph and its titles. The main drawback to using a bitmap image is the significant loss of resolution that occurs when the program importing the graph stretches, shrinks, or otherwise scales the image. This is especially true for an aspect ratio that differs significantly from the original image.

You will almost always be successful pasting a clipboard bitmap into an application. Although the Device Independent Bitmap name suggests a similar success in importing these DIB files, Windows applications are not consistent in their handling of DIBs. You may find wide variations in DIB handling even amongst different software applications from a single company. TableCurve 2D always writes the common 8-bit 256 color DIB, even if you are using a different video resolution.

Metafiles

Windows metafiles consist of a series of Windows graphics vector instructions which are intended to reproduce the original image. Windows Metafiles offer superior resolution and scaling, although the program importing the image may have difficulties in properly rendering text elements such as superscripts, subscripts, and symbols in their proper positions.

Metafiles are highly sensitive to the individual applications importing them. If you are unsuccessful in

pastings a clipboard metafile image into your software, first check the Windows Clipboard Viewer to insure the image is actually there. If so, and if your software supports WMF file import, you should try importing a disk-based metafile. The [Windows Metafile to File in Aldus Placeable Format](#) option produces a disk-based Windows Metafile that is "placeable" and which is based upon the Aldus specification. This is the Windows Metafile format preferred and sometimes required by high-end desktop publishing and drawing software. The default file extension for Metafiles is [.WMF].

You may also wish to try the 32-bit Enhanced Metafile formats if your importing application is a 32-bit Win32 program which supports this new format.

Windows has traditionally offered no intrinsic support for subscripts, superscripts, or changing to a symbol font in the middle of a text string. As such, a title containing formatted text has required multiple text output instructions. When a metafile image is placed and scaled, the software importing the file may not preserve the proper character spacing across the separate text segments. If this occurs, try initially placing the image in the aspect ratio of your video display. If this fails, you should uncheck the [Preserve Superscripts, Subscripts, and Symbols in Metafiles](#) box.

Color and Bounding Frame

To copy the graph as a Page-White image, simply uncheck the [Use Color](#) item. When the [Bounding Frame](#) box is checked, there will be a thin-box at the boundaries of the image.



Graph Custom Titles

The **Basic XY Titles**, which consist of a main, X, and Y title, are used within the default titles provided in the various graphs. The **Custom Titles** option offers the means to extensively customize the titles used within a **TableCurve 2D Graph**.



Note that titles and labels can be toggled off in the **2D View** option. They need not be cleared in this custom titles option.

This option offers the means to add or edit up to five titles as well as the X, Y, and Y2 titles. This option also offers the means to control the sizes of these titles. You will need to use this option if you wish to add superscripts, subscripts, or special upper ASCII or symbol characters to the titles. This option offers the means to save the custom titles to disk for subsequent use.

Titles Entry

You may enter up to 5 titles for display above the graph. The titles presented will either be the default titles for the graph or the custom titles currently active. To reset the defaults, use the **Reset** button. If you will be reading in a custom title file, you may wish to use the Cut or Copy button to place one of the titles in the clipboard before the previous titles are read. You would then use the Paste button to paste this over the previous title not pertinent to the present graph.

Subscripts, Superscripts, Bold, Italic

To create a subscript, superscript, or to make a portion of a title bold or italic, first enter the text and then highlight the portion of text you wish to superscript or subscript. When you press the subscript, superscript, bold, or italic button, the appropriate control codes are inserted into the text. Nested superscripts or subscripts are ignored.

Special Characters

If you wish to add a character from the upper 128 characters of the current font, or from the Symbol font, place the cursor at the desired location where you wish to have the special character inserted. When you press the Symbol (μ) button, you will be presented with a display of the available characters. Select the character desired and it will be inserted, along with the appropriate control codes, into the title text.

~ Character

The ~ character is used internally by TableCurve 2D for formatting subscripts, superscripts, and special characters. In order to use the ~ character in a title, you must enter the ~ character twice.

Sizes

You may set the font size of the main title, the secondary title, the third-fifth titles, the X,Y,Y2 titles, and the X,Y labels. The font sizes appear to the right of the titles. Simply enter the point sizes desired, or set the values using the spin buttons, or select the values from the popup menu activated by right clicking and holding the mouse on the field or spin buttons.

Titles will not appear when the **Graph Layout** is maximized. If a fixed size layout is selected, one or more of the titles may not be displayed if the font sizes are too large.

Saving Titles

All of the information in the Custom Titles screen can be saved to disk. These files have default TTL extensions. These are binary files which can only be generated from within the program. Custom titles are not saved across sessions. You must explicitly [Save](#) a set of custom titles if you wish to use them again.

Recalling Titles

The [Read](#) item will read previously saved custom title information into the current graph. For curve-fit and residuals graphs, the equation, goodness of fit statistics, and coefficient titles will be automatically updated if they were originally saved without modification. If you modify and save custom titles containing equation information, these titles may not be updated and could contain incorrect information relative to the present equation. If the second title begins with "[Rank](#)" it will be automatically updated for the current equation. If the third title begins with "[r2=](#)", it will be updated. If the fourth title begins with "[a=](#)", the fourth and fifth titles will be automatically updated. If you wish to use one of the default titles for the current graph in place of a title that will be read, use Cut or Copy to place that title in the clipboard, read the title file, and then paste it into or over the title not pertinent to the current data or fit.

Scanning Curve-Fit Equations

Custom titles are preserved when scanning curve-fit graphs. Titles for the current equation, its goodness of fit statistics, and its coefficients will automatically replace the corresponding custom titles if the second custom title begins with "[Rank](#)", the third custom title begins with "[r2=](#)", and the fourth custom title begins with "[a=](#)". If you modify the initial characters of these titles, they will be considered constant strings and as such equation information contained within them will not reflect the current equation being scanned.



Graph Point Format

The **Data Points** dialog for a **TableCurve 2D Graph** itself incorporates a copy of the current graph. All changes made in any of the point format options are immediately reflected in the graph. The entry fields serve either the Y or Y2 axis, depending on whether **Y Axis** or **Y2 Axis** is selected. The Y2 options will not be available if the graph contains a single plot. This option sets only the principal data points of each Y axis. Reference data sets, if present, use a preset format.

Size

A graph's data point size may be set from zero to ten. Size 0 points are always pixels (non-bar shapes) or lines (bar shapes), regardless of the device. If the graph specifies a bar shape, this option affects the width of the bar.

Shape

A graph's points may be set as square, circle, diamond, triangle down, triangle up, or plus symbols. You may also choose a bar graph where the bar's base is zero, or the Ymin or Ymax of the plot.

Fill

You may choose to leave the points unfilled or fill them with one of the four point colors within TableCurve 2D.

For **Review Graphs**, you may also choose **Color by Residuals**. Here the points are colored based upon the absolute value of the number of standard errors from the Y-predicted value. The default color schemes employ a coloring based upon ascending wavelength. Points less than 1 standard error are shown in blue, those between 1 and 2 standard errors are green, those between 2 and 3 standard errors are yellow, and those beyond three standard errors are red. Points that are beyond 2 standard errors may represent outliers which adversely impact the overall fit.

Options

- You may elect to have inactive points filled or unfilled. To differentiate inactive points on black and white printers, you will need to use unfilled inactive points.
- Points may be hidden entirely.
- A graph's points can be connected using lines.
- A graph's points can be connected using a **cubic spline**
- A graph's points can be connected using a **$\chi\upsilon\beta\iota\chi$ NYPB Σ** (non-uniform rational B-spline)



Graph Resolution

TableCurve 2D Graphs can render functions at various levels of resolution. Full computes functions at every pixel to be displayed on screen and on every dot in the printed output. 1/2 computes functions at every other pixel. 1/10 computes functions at every tenth pixel.

Unless you are encountering unsatisfactory rendering or printing performance, the Full option is recommended since it offers the greatest likelihood for detecting any kind of anomaly within the fitted curve.

Graph Scaling

The **Scaling** dialog for a **TableCurve 2D Graph** itself incorporates a copy of the current graph. All changes made in any of the scaling options are immediately reflected in the graph.

This custom scaling option is used for special scaling needs. For zooming in any given graph, it is much easier to use the left mouse button, starting at one corner of the desired region and pressing and holding the button while moving the mouse to the opposite corner of the desired region. The zoom-in can then be cleared simply by right clicking the mouse anywhere within the graph region, except upon a point or function anchor.

The X, Y, and Y2 (if present) axes are treated independently. The Y entry fields serve either the Y or Y2 axis, depending on whether **Y** or **Y2** is selected. The Y2 options will not be available if the graph contains a single plot.

The **Automatic** option produces TableCurve 2D's default scaling for the particular axis. To manually scale an axis, this Automatic box must be unchecked.

The **Log** option will produce a logarithmic axis. No errors are reported for any data or functions which have negative values. The points or function simply will not be drawn. Some graphs, such as **Residuals Graphs**, use an absolute value when a log scale is used.

The **Reverse** option creates a descending rather than ascending axis. This may be of value in spectral plots where the X axis is Wave Number.

The **Common Y,Y2** option specifies a common axis be used for both Y and Y2 plots. The ranges are simply adjusted to accommodate all information in both plots.

The **Priority** option allows you to have Y axes scaled for either **Data** or **Fn** (Functions). With a Data priority, the minimum and maximum in the primary data determine the scaling. With a Fn priority, the minimum and maximum of all functions plotted determine the scaling limits. If a user function is wholly out of control (nowhere to be seen amongst the data), changing from a Data to Fn priority may be of value.

The **Zoom** option enables an axis to be automatically zoomed out in ten increments. These steps will be linear for linear scaling. When the scale is logarithmic, alternating log units are incrementally added to each side of the scale.

The **Min**, **Max**, and **Divs** fields are active when manual scaling is used (Automatic is off). The values you enter are used exactly as the limits of the scale. The divisions field specifies the number of divisions, not the number of interior grid positions (the interior grid count will be one less than the number of divisions). For logarithmic scaling, you are free to set non-integer powers of 10, and subsidiary logarithmic grids will be drawn, although they will lack the traditional meaning.



Grids are specified in the **Graph View** option.

Reset

The **Reset** option restores default scaling.



Default Scaling can also be reset within a **TableCurve 2D Graph** by clicking the Reset Default Scaling button.

Save

The [Save](#) option will save a binary custom scaling [SCL] file to disk for future recall and use. You should save any custom scaling that you may wish to use for a number of replicate or similar data sets.

Read

The [Read](#) option imports a binary custom scaling [SCL] file previously saved.

To accept the custom scaling shown in the dialog and graph, simply click on [OK](#). To abandon any modifications, click on [Cancel](#).

Printing Graphs

TableCurve 2D Graphs offer a full print preview [Print](#) option. You can use this option to print a half-page, full-page, or custom size graph.

The dialog will contain a preview image of the graph, rendering the image to be printed as accurately as possible. This preview graph will be updated as orientations are changed, options are selected, margins are set, and font scalings changed.

Frame Graph

You may elect to print a bounding frame around the graph by checking the [Frame Graph](#) option.

Color or Black and White

You may print the graph in color by checking the [Use Color](#) item. Note that black and white graphs can be printed regardless of the current [color scheme](#). TableCurve 2D internally uses the Page White BW color scheme for monochrome printing.

Adding Numeric Summary

This option is available in both fixed and custom portrait modes. The numeric summary is printed immediately below the graph. This option is available only for [Curve-Fit Graphs](#).

Margins

If you wish to print a custom portrait or landscape graph, you must specify the four margins which define the boundaries of the graph.

Font Scaling

In some instances, TableCurve 2D may use a larger screen font to maintain legibility in graphs. If this Print Preview indicates the need to increase or decrease the overall font scaling, you may enter a different value in the [Font %](#) field. You may also use the spin buttons to change this value or you may select the value from a popup menu activated by holding the right mouse button down when over the field or spin buttons.

Line Width

With high resolution output devices, the single width lines used for drawing peaks may appear too light. These lines can be darkened by increasing the line width. Since this width cannot be effectively represented within the limited screen resolution, the print preview will not reflect this line width setting.

Save Custom Settings

You may [Save](#) a custom print file containing the current information. This custom print graph file will have a default [CPG] extension.

Read Custom Settings

Use the [Read](#) option to import a custom print settings from a custom print [CPG] file.

Printer Setup

To select a given printer, use the **Setup** button. When you select a printer, you will also have the option to modify the printer driver's own dialog, as in the case of setting printer resolution, paper size, and printing multiple copies.

Initiating the Print

Simply press **OK** to begin the printing, or **Cancel** to abandon printing. Once printing begins, there will be a brief time where a print job can be canceled during the Windows spooling operation. After that time, a print job must be canceled from Windows Print Manager.

Postscript Printing

When printing, the print spooler within Windows writes a file to its temporary directory. When printing a TableCurve 2D graph to a Postscript printer, this file can be as large as 3-4 Mb in size. A Windows error message will result if there is insufficient space on the temporary drive. If this occurs, the best solution is to make sufficient disk space available, by freeing space on the current drive. As an alternative, you may specify a reduced **Function Resolution** for the graph. This can significantly reduce the size of Postscript streams.

Discerning Inactive Points on Black and White Printers

The default inactive point state uses a different color rather than an unfilled symbol. To discern inactive points on black and white printers, you will need to uncheck the **Inactive Fill** in the **Point Format** dialog.

Driver Problems

For some printer and plotter drivers, particularly those from HP, you may need to set the orientation for the plot within the printer setup dialog.

You may also find that the HP Plotter driver does not support rotation for True Type and Adobe Type Manager fonts. You may need to use the Modern, Roman, or MS San Serif font in order to have the proper rotation for the the Y-axis title.

TableCurve 2D Graphs use a special procedure for producing enhanced grid lines which look very good with digital devices. For plotter output, you will probably need to set a Grid Level of 0 (a pure line) in order for grids to appear. This zero grid level may also be necessary with certain versions of the Deskjet driver which only partially render grid lines or which do so with incorrect colors.

ASCII Text Editor

The TableCurve [ASCII Editor](#) is a NotePad-like editor available in many of the program's menus. The [ASCII Editor](#) option in the Edit menu within the program's main menu is used primarily for editing the X-Y data table since it offers an update of the X-Y data table with the numeric contents when closing the editor. Most of TableCurve's principal text windows offer an ASCII Editor option which will place the contents of the window into the editor. If you wish to edit a non data table text file from the main menu, you may select the [ASCII List](#) option in the main File menu, select the file desired, and then select the ASCII Editor option from the list window's Edit menu.

Considerations

- There are no size limitations if you are running Windows NT or Windows 2000.
- For Windows 95 and 98, which have a 16-bit subsystem, edit controls remain limited to 64K. In this case, when reading files larger than 32K, the ASCII Editor automatically strips all whitespace except that necessary to delimit the information. When editing a large equation list, the information will be truncated to fit within the editor, allowing an edit of the first 700-800 equations.
- Columnar formatting, if present, can be preserved in the editor using a fixed width font, such as Courier. Special characters such as subscripts, superscripts, and symbols will not be preserved.

File Menu

The standard items to create a New file, to Open an existing file, to Save the current file, to Save as a different file, and to exit are here. The contents of the editor may be saved as an ASCII text file, a Lotus WK1 file, or a Lotus WK3 file. Although a WK1 or WK3 can be created regardless of the editor contents, such files generally make sense only for columns of numeric information.

The contents of the ASCII Editor can also be [directly streamed to MS Word](#).

The [Printer Setup](#) item is used to select and optionally configure a printer for use by TableCurve. Note that the orientation option in the printer driver's own configuration dialog will be overridden by the orientation set in the Text Print dialog.

The [Print](#) item opens the [Text Print](#) dialog, allowing a header to be specified, the date and page number to be included, the orientation to be chosen, and the margins to be set. When printing from the editor, it is not possible to center the longest line nor to print only the visible portion of the editor.

Edit Menu

The standard items to [Cut](#) to, [Copy](#) to, and [Paste](#) from the clipboard are here as well as a single step [Undo](#), the [Delete](#) of selected text, and a [Select All](#) for selecting all text. The [Copy](#) option will copy the selected or highlighted text to the clipboard in a text format, and if at least a full line is selected, also in a WK1 spreadsheet format suitable for a very rapid paste by Excel, Lotus, or Quattro Pro.

Search Menu

The standard items consist of a [Find](#), a [Find and Replace](#), and a [Next](#) Item which will continue an existing Find or Find and Replace operation.

Style Menu

The editor always loads with a standard Courier font. The [Font Select](#) option will allow any other font to be used. For editing the content of TableCurve text windows, a fixed width font is recommended.

There are four formatting items in the Style menu. The [Comma to Decimal](#) item is for converting European format numeric entries to the decimal type required by TableCurve. This option simply converts all comma characters within the file to the period character, as in [123,45] being converted to [123.45]. You may enter data with commas as decimal separators and then use the Comma To Decimal option to convert to it to the format required by TableCurve.

The [Space Delimited](#) option will strip all whitespace from the program except for a single space delimiter between the entries. The [Comma Delimited](#) option accomplishes the same task, except that the single delimiter will be the comma character. In the [Tab Delimited](#) option, the tab character serves as the delimiter.

Text Viewer

The TableCurve text window appears throughout the program for displaying, modifying, and printing text information.

Window Type

Text can be displayed either as RTF (Rich Text Format) windows or in a fast standard display.

If [Rich Text Format Windows](#) is checked in the [Preferences](#), RTF-type windows are used to display reports. RTF windows offer the means to select and copy portions of the overall text, but are considerably slower than TableCurve 2D's standard text windows. For windows whose contents vary based upon the size of a data set, RTF windows are available only for data sets having 128 or fewer points. Data tables larger than 128 points will be displayed using standard text windows regardless of this setting.

The standard text windows can display up to 65536 lines of information. You may use either the cursor keys or the scroll bars to move about the text. If the contents of the window have been generated by TableCurve 2D, the text will contain columnar formatting to properly align all fonts, there will be a predefined color formatting, and there will be support for displaying and printing subscripts, superscripts, and symbols.

Editing the Content of TableCurve 2D Reports

TableCurve 2D directly supports only non-formatted [ASCII editing](#) of report information. To edit the formatted report as it appears in TableCurve 2D, you must save the contents to an RTF file for import into a word processor, copy the contents to the clipboard and then paste them into an RTF-capable editor, or you can [stream the content directly to MS Word](#).

File Menu

The viewer's File menu contains options to save the information in the window as an ASCII text file, as a WK1 or WK3 worksheet, as a fully formatted Rich Text Format (RTF) file, or the contents can be [directly streamed to MS Word](#). Although any text can be converted into a WK1 or WK3 spreadsheet, the primary use will be with numeric columns of data.

The RTF format preserves the appearance of the text when imported into word processing and desktop publishing products. The RTF option thus preserves columnar formatting, font colors, bolding, superscripts, and subscripts. In order to save, copy, or stream black and white RTF text, you will need to uncheck the [Color](#) option in the Style menu.

The [Printer Setup](#) item is used to select and optionally configure a printer for use by TableCurve 2D. Note that the orientation option in the printer driver's own configuration dialog will be overridden by the orientation set in the Text Print dialog.

The [Print](#) item opens the [Text Print](#) dialog, allowing a header to be specified, the date and page number to be included, the orientation to be chosen, and the margins to be set. You may also elect to center the longest line and to print either all information or only that portion currently visible in the window.

Edit Menu

The [Copy](#) option copies the text in the window to the Windows clipboard as unformatted ASCII text, as formatted Rich Text Format (RTF) text, and in the Lotus WK1 format. The unformatted text format is space delimited and is used when you to paste the information into a text-based program.

Again, since the RTF format preserves columnar formatting, font colors, bolding, superscripts, and subscripts. In order to save a black and white RTF text file, you will need to uncheck the [Color](#) option in the Style menu prior to the Copy.

For some products where RTF is the default clipboard format, such as Microsoft Word, you will need to use a Paste As, Paste Special, or similar option in order to specifically select the unformatted text. The most practical use of the WK1 spreadsheet format will be with text containing columns of numeric information. This Copy option is limited only by available memory.

The [ASCII Editor](#) option copies the contents of the viewer into the TableCurve ASCII editor. On 16-bit Windows subsystems (Windows 95 and Windows 98), the contents of must be less than 64K in size in order for the ASCII Editor to be available.

Style Menu

The text windows each use their own specific font. If you select a different font using the [Font Select](#) item, it will be used only for the current type of text window. Since TableCurve 2D's internal windows are preformatted, you may freely select variable pitch fonts, although you may wish to use a fixed width font for the ASCII List and ASCII Editor options, especially when working with files containing numeric columns of data.

The [Color](#) item simply toggles the text between its predefined color format and the default black text on white. The font and color setting are saved across sessions for each type of window.

Printing Text

The [Print](#) option is available in all of TableCurve 2D's principal text display windows.

You can print all of the information in any given window or only that portion currently visible on the display.

You may elect to print a special header on each page, and to have the data and page number printed. The header can be multi-line, although it is limited to a total of 1024 characters.

You may choose to print the text flush left or in such a way that the longest line is centered.

You may optionally adjust the four margins which define the boundaries of the printed text.

The information can be printed in color or in monochrome.

You may save a custom print file containing the current information or read such a file to set custom print criteria. This custom print text file will have a default [CPT] extension.

To select a given printer, use the [Setup](#) button or the [Printer Setup](#) menu item. When you select a printer, you will also have the option to modify the printer driver's own dialog, as in the case of printing multiple copies.

Note that the orientation option in the Print Text dialog will usually supersede the orientation set in the printer driver's own configuration dialog. For some output drivers, such as HP Plotters, it may be necessary to set the orientation in the printer driver's own dialog.

TC2D Automation

Most algorithms within TableCurve 2D offer a powerful automation facility that enables the automatic processing of large numbers of data sets in the same manner as the one currently being processed.

Data Source

The data source can be one of the following:

- Data streams returned using a TableCurve 2D format automation DLL
- Data sets stored in an Excel 95 or Excel 97/2000 format file

Automation DLL



TableCurve 2D offers the means to acquire data using a DLL interface. The program ships with [tcauto1.dll](#), [tcauto2.dll](#), and [autoacq.dll](#), sample DLLs that simulate different data acquisitions. These DLLs and their C/C++ source code is in the Automation subdirectory. These files should be used as templates for creating automation DLLs.

TableCurve 2D will optionally pass a setup value from 0 to 255, allowing up to 256 acquisition schemes, sampling, and so forth from a single DLL.

Each data stream acquired can be up to 65536 points in length.

The [Test Automation DLL](#) option in the Help menu can be used with your automation DLL to generate Excel worksheets.

Excel 2000/97/95 File



If the data source is an Excel 95 or 97/2000 file, you must specify the file, and whether to process the first sheet, all sheets, or a specified sheet. You must also specify how the data are stored. Two formats are supported. For [XY Pairs](#), the first X must be in column A, the first Y in column B, the second X in column C, the second Y in column D, and so on. For [Single X](#), the X values must be in column A, the first Y in column B, the second Y in column C, and so forth. Typically the main data titles will be in the first row, the X and Y titles in the second row, and the data will begin in the third row.

Excel 95 is limited to 16384 rows while Excel 97/2000 supports up to 65536 rows. If the first two rows are used for titles, the data length limits will be 16382 and 65534.

The TableCurve 2D main window displays the amount of free RAM available. If this quantity is sufficient, TableCurve 2D will load the full Excel file into memory, resulting in very fast import of each successive data set. If the free RAM is insufficient, TableCurve 2D will need to read each set from disk, a significantly slower operation, especially with large files.

Be sure the Excel file you are processing is not currently open within Excel. This not only consumes memory, but requires that TableCurve 2D make a temporary copy of the file on disk before each import.

TableCurve 2D reads Excel files directly, without using OLE automation. There is no need for MS Excel to be open, and TableCurve 2D does not launch it.

The [Test Automation DLL](#) option in the Help menu can be used with one of the sample DLLs (or your own automation DLL) to generate large Excel worksheets for testing automation performance.

Processing

The graph(s) for the current procedure are updated as the data sets are imported and processed.

Numeric information fields are also updated. This update occurs even when no export options are selected.



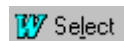
When the processing is underway, all options are disabled except the [Reset](#) button. The automation can be halted by clicking this button or by pressing the Space bar, Enter, Tab, End, or Esc keys.

When the automation ends as a consequence of user intervention, an option is presented to transfer the current data set into the program's XY data table. This may be useful when screening large numbers of data sets for an anomalous condition. In general though, such screening procedures are done in an unattended manner using MS Word and/or MS Excel export.

Export

Two export options are generally available. Either or both can be selected. The first is an MS Word/RTF (Rich Text Format) file containing numeric summaries and/or graphs. The other is an Excel 95 or Excel 97/2000 file containing numeric data.

MS Word/RTF Export



MS Word/RTF export is available for all options where graphs and reports are useful. If this export is available, there is a choice to [stream the output directly to MS Word](#) (Word 95, Word 97, and Word 2000 are supported), or you may write the information to a portable RTF (Rich Text Format) file.

The content available for export varies widely with the procedure. In most procedures, at least one type of graph can be selected. When an option employs dual graphs, both will be automatically written to the file. The graphs will be constructed using the current plot settings and sizes, and inserted as Windows Metafiles. You can select landscape or portrait mode for the graphs. The dimensions of the graph within the file will be based upon 96 pixels per inch. Note that beyond a certain size, the graphs utilize full landscape or portrait pages.

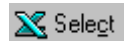
Numeric reports are included using the current options set for them. For RTF files, you can independently select landscape or portrait mode for the inserted text. For [MS Word streamed export](#), a single orientation applies to both graphs and text.

Note that RTF files with graphs can grow quite large when processing an appreciable number of data sets. Be certain you have sufficient hard disk space on the drive where the exported RTF file is being written before initiating an automation session. An RTF file is written directly to disk, without using OLE automation. There is no need for MS Word to be open, and TableCurve 2D does not launch it.

MS Word DOC files are more efficient than RTF files in storing graphs. This is particularly true of the newest releases which compress metafile information.

The general [Preferences](#) option contains the [Use A4 as Default Paper Size](#) option. By default, MS Word and RTF export is formatted in 8.5 x 11 size. If you want the export formatted for an A4 paper size, you must check this preference. The A4 preference also applies to the [MS Word and RTF export](#) option.

Excel Export



All procedures supporting automation offer Excel export. Some offer extended numeric data while others write only basic two column data. If [Extended Data](#) is available, you may elect to disable it and write only the basic data. Extended data is written to the Excel file in the format used for displaying information in the List Data option of the procedure. For the Curve-Fit Review, the export is only available in an extended format.

The export can be written using one worksheet for each data set, packing as many data sets as possible

per worksheet, or if possible, the export will be written to match the input Excel format. Both Excel 95 and Excel 97/2000 formats are supported. Excel 95 is limited to 16384 rows while Excel 97/2000's limit is 65536 rows. TableCurve 2D limits the number of sheets written to 255. The automation will be automatically closed if this sheet limit is reached.

The last two columns of each worksheet are not used, in case you wish to average data across all columns on a given sheet. Similarly, the 256th sheet is reserved in case you want to average data across sheets.

Just as with the RTF export, Excel files can grow quite large, especially when large amounts of numeric information are written for each data set. Again, be certain you have sufficient hard disk space on the drive where the exported Excel file is being written before initiating an automation session.

The Excel file is not immediately written. For optimum real-time performance, the information is first written in a streamlined binary format that uses very little processing time. When the automation is finished, this temporary binary file is converted to an actual Excel file. Although the Excel file is written directly, without OLE automation, this may still require some time if very large files are involved. You may work within Excel while an automation session is underway.

Background Automation

TableCurve 2D's automation is designed primarily for unattended processing of large numbers of data sets. To run an automation session in the background, while working on foreground tasks in other applications, simply minimize TableCurve 2D's current or main window. In Windows NT and Windows 2000, once an automation session is underway, you may want to launch the Task Manager, right click on TableCurve 2D, and set the process priority to low. This should maximize foreground performance.

Keep in mind that the Excel data source automation will be most efficient when sufficient RAM exists to load the full Excel file into memory. Many applications, such as MS Word and MS Excel, consume large amounts of RAM. Windows itself consumes a surprisingly large quantity of RAM. Typically, unless you have 64 Mb of RAM or higher, Excel data source automation will suffer if the Excel file is large and other major applications are running concurrently.

Test Automation DLL

The [Test Automation DLL](#) option in the Help menu is provided to test automation DLLs. Data is imported from the DLL and written directly to an Excel file for inspection.

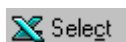
Automation DLL



This button is used to select the automation DLL file.

TableCurve 2D will optionally pass a setup value from 0 to 255, allowing up to 256 acquisition schemes, sampling, and so forth from a single DLL. This [Optional Setup](#) value need not be used. Each data stream acquired can be up to 65536 elements in length.

Excel Export



This option produces Excel export for the basic data. The export can be written using [1](#) worksheet for each data set or as many data sets as possible can be packed per worksheet ([Maximize](#)). Both Excel [95](#) and [97/2000](#) formats are supported. Excel 95 is limited to 16384 rows while Excel 97/2000's limit is 65536 rows. With two titles exported, data length limits will be 16382 and 65534. TableCurve 2D limits the number of sheets written to 255. The automation will be automatically closed if this sheet limit is reached.

The last two columns of each worksheet are not used, to support averaging data across all columns on a given sheet. Similarly, the 256th sheet is reserved for averaging data across sheets.

Since no processing is done with this option, the Excel files written in this test option can grow quite large very quickly if data are continuously available.

The Excel file is not immediately written. For optimum real-time performance, the information is first written in a streamlined binary format that uses very little processing time. When the automation is finished, this temporary binary file is converted to an actual Excel file. Although the Excel file is written directly, without OLE automation, this may still require some time if very large files are involved.

MS Word/RTF File Export

This option is used to export graphs and reports to Microsoft Word and other word processors.

Destination

The output can be streamed directly to Word 95, Word 97, or Word 2000 by checking the [Stream to MS Word](#) box. This option will not be available if MS Word is not installed on your system. This option automatically opens Word and places the specified content within the document that is created.

The [RTF File](#) option specifies that a Rich Text Format file be written to disk. This format can be imported by all major word processors and by Wordpad, an RTF editor that ships with Windows.

Content

Each procedure will have its own content items. Select the items you wish to include in the export. When multiple items are available, the sequence of exported items is fixed.

Options

The [Landscape Text](#) and [Landscape Graphs](#) checkboxes are used to specify landscape rather than portrait mode for numeric reports and graphs. If the [Stream to MS Word](#) option is used, text and graphs must share the same orientation. Mixed portrait and landscape mode is available only for RTF file export.

The [Maximize Size of Graphs](#) option will automatically use the full available page area. For landscape mode this is the entire page. For portrait mode, this uses approximately half of the page. When this option is unchecked, the size of graphs will be proportional to the size of their windows in TableCurve 2D.

The [Insert Page Breaks](#) separates each report or graph with a page break.

The [Save After Each Stream Inserted](#) option will save the Word document after each item is inserted.

The general [Preferences](#) option contains the [Use A4 as Default Paper Size](#) option. By default, MS Word and RTF export is formatted in 8.5 x 11 size. If you want the export formatted for an A4 paper size, you must check this preference. The A4 preference applies to this MS Word and RTF export option and also to all reports generated using the [automation](#) facility.

Stream to MS Word

TableCurve 2D offers a means to stream all information in reports and graphs to Microsoft Word. Word 95, Word 98, and Word 2000 are supported. This option automatically opens a file in MS Word and places reports and graphs directly in the document.

Considerations

- Once a stream is opened by specifying a file and having Word automatically opened, the stream remains open so long as this file has the active focus within Word. This open file remains the stream target until Word is closed, until the file is closed in Word, or until another file is made active within Word.
- The connection between TableCurve 2D and Word is usually broken in Word, not in TableCurve 2D. In Word, you must close the target file for the stream, change to a different file, or close this instance of Word. The connection is broken by TableCurve 2D only when it is closed.
- The stream source is not saved across TableCurve 2D sessions. If TableCurve 2D is closed and restarted, a new Word file must be specified and a new instance of MS Word is launched.
- Word locks all files being edited for its exclusive use. You cannot specify a file as a stream target if it is already open in Word. To append to an existing file, you must first close this file within Word. It can then be specified as the stream target.
- When the Word file is first specified, there is an option to overwrite an existing file or to append streamed information to the existing content.
- You can freely edit the contents of the information within Word while the connection between TableCurve 2D and Word remains in place. New information is always appended to the end of the Word file.
- This procedure utilizes the Windows clipboard, replacing existing clipboard content with the stream content.

Text Windows

The contents of text windows in TableCurve 2D are directly streamed to Word when the [Stream to MS Word](#) option is used from the text window's File menu. Except for the first-time specification of a file, the only evidence of this transfer will be in Word.

Formatting, if present, is automatically preserved and carried over to Word. For Black and White text, uncheck the [Color](#) option in the Style menu.

The Word file is automatically saved after each stream has been sent.

Least Squares Curve-Fitting

Linear Equations

A **linear equation** is one whose coefficients appear linearly in the expression. For example, the following equation is linear in its coefficients:

$$y = a + bx + cx^2 + dx^3$$

Here the coefficients a, b, c, and d linearly multiply the 1, x, x², x³ basis functions of x. Such linear equations can be solved in a single step matrix solution.

Note that linear equations refer to the relationship between the coefficients and basis functions, not to the graphical appearance of the curve. A curve may have a wildly non-linear appearance, but yet be formed by a linear equation.

Non-Linear Equations

A **non-linear equation**, on the other hand, is one where the coefficients appear in a non-linear or nested fashion. The following equation is linear only in the a and b coefficients, and non-linear in c and d:

$$y = a + b \exp \left[-\frac{1}{2} \left(\frac{x - c}{d} \right)^2 \right]$$

A non-linear equation cannot be solved in a single step. Estimates must be made for each coefficient, and some form of iterative convergence must follow.

Minimization of Least Squares

In TableCurve 2D, the fitting of linear equations always minimizes the sum of squares of the residuals. A residual is defined as:

$$y_i - \hat{y}_i \text{ where } \hat{y}_i = f(x_i, a, b, c, \dots)$$

A residual is simply the difference between the y value of a given x,y data pair and the y value computed from the curve-fit equation at this same x value. A residual is thus the vertical y-distance between the curve and a data point. It can be either positive or negative in value. The square of a residual is always positive, and thus reflects the magnitude of the residual, although it does so in a second order rather than a linear fashion.

Least squares minimization thus assumes that the x values are accurately determined, and that an error exists only in the dependent variable y. In doing a least squares minimization, one assumes that the errors map to a Gaussian profile. Such errors are said to be normally distributed.

Design Matrix

In a linear equation it is possible to construct a design matrix based upon the sums of the basis functions and their cross-products. Let us consider the simple equation for a quadratic:

$$y = f(x, a, b, c) = a + bx + cx^2$$

The first step is to construct the least squares merit function. This is often referenced as the chi-square:

$$X^2 = \sum_{i=1}^n w_i (y_i - f(x_i, a, b, c))^2$$

This is simply the weighted sum of squared residuals. The goal of all least squares curve-fitting is to minimize this chi-squared function. To do this, we must take its partial derivatives with respect to each of the parameters:

$$\frac{\partial y}{\partial a} = \sum_{i=1}^n -2w_i (y_i - f(x_i, a, b, c)) \frac{\partial}{\partial a} f(x_i, a, b, c)$$

$$\frac{\partial y}{\partial b} = \sum_{i=1}^n -2w_i (y_i - f(x_i, a, b, c)) \frac{\partial}{\partial b} f(x_i, a, b, c)$$

$$\frac{\partial y}{\partial c} = \sum_{i=1}^n -2w_i (y_i - f(x_i, a, b, c)) \frac{\partial}{\partial c} f(x_i, a, b, c)$$

The chi-squared will be at a minimum when these three partial derivatives are zero. By setting these three equations to zero and taking the partial derivatives of this quadratic equation, the following design matrix and constant vector is produced:

$$\begin{vmatrix} n & \sum_{i=1}^n w_i x_i & \sum_{i=1}^n w_i x_i^2 \\ \sum_{i=1}^n w_i x_i & \sum_{i=1}^n w_i x_i^2 & \sum_{i=1}^n w_i x_i^3 \\ \sum_{i=1}^n w_i x_i^2 & \sum_{i=1}^n w_i x_i^3 & \sum_{i=1}^n w_i x_i^4 \end{vmatrix} \begin{vmatrix} \sum_{i=1}^n w_i y_i \\ \sum_{i=1}^n w_i x_i y_i \\ \sum_{i=1}^n w_i x_i^2 y_i \end{vmatrix}$$

Note that the weights w_i are inverse variances or $1/\sigma_i^2$, where σ_i is the standard deviation for a given data pair. TableCurve 2D assumes w_i 's of 1.0 unless specific weights are entered.

The determination of a , b , and c requires no more than the computation of a handful of sums and the solution of the three simultaneous equations, a very fast procedure compared to the iterative procedure required for non-linear equations.

Non-Linear Least Squares Fitting

When an equation is non-linear, the least squares procedure is similar. In this instance, however, it is not possible to express partial derivatives as a function of only the x basis functions. In a non-linear equation, the partial derivative with respect to one coefficient will include one or more of the other coefficients,

making a single step matrix solution for the individual coefficients impossible.

As such, non-linear fitting consists of an iterative procedure that begins with an initial set of estimates for the parameters. Although there are a variety of methods to converge to the minimum least squares solution, all procedures must compute a point by point sum of squared residuals for each iteration's set of coefficients. This is especially time consuming with large data sets.

TableCurve 2D uses the Levenburg-Marquardt algorithm for fitting its non-linear equations and user-defined functions. Although this algorithm requires a matrix inversion and the computation of partial derivatives for each iteration, its rate of convergence is among the best of available methods.

Global vs. Local Minimum

Unlike linear fitting, non-linear fitting does not guarantee the minimum least squares solution. Non-linear fitting algorithms sometimes find a local minimum in the n-dimensional space of the fit rather than the true global minimum which represents the desired least squares solution. Good starting estimates are thus essential. TableCurve 2D determines effective starting estimates for the parameters in all of its built-in non-linear equations during its pre-scan procedure of the input data. For UDF's, TableCurve 2D offers the means to graphically adjust UDF estimates prior to fitting. Accurate starting estimates insure convergence to this global minimum.

No Exact Solution

An exact least squares solution on a digital computer would involve minimizing the sum of squared residuals to the full floating point precision of the machine. For Intel and compatible coprocessors, this would mean to 19 digits of precision.

A non-linear fit is sometimes referenced as asymptotic, meaning it produces a coefficient vector that only approaches the limit of exactly minimizing the sum of squared residuals. This is because you iterate to a specified tolerance or fractional error in the minimization. This tolerance is always less than the floating point precision of your computer.

TableCurve 2D's default non-linear fractional convergence is 1E-6, but can be set anywhere from 1E-3 to 1E-15. The 1E-6 value means that the **r2 goodness of fit** must be unchanging through six significant figures for five iterations before the algorithm signals convergence.

The one-step linear minimization is likewise likely to produce less than the 1E-19 floating point granularity of the FPU, especially for higher term count equations. For example, consider the following TableCurve 2D polynomial:

$$y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10}$$

The design matrix for this equation will contain elements that vary widely in magnitude. For example, the weighted n might be 100 whereas the weighted sum of x20 might be 1E30. The problem arises due to the limitations of finite floating point in matrix operations where the matrix is close to singular.

A nearly singular matrix has a determinant that is nearly zero. In such cases it is difficult for conventional matrix solution methods to produce an accurate least squares minimization.

One solution to this problem is to use higher precision math than that offered by the hardware floating point unit. This approach is implemented in the **high precision polynomials and rationals**.

Another solution is the **SVD algorithm**.

Singular Value Decomposition

When working with digital computers, no computation can be any more precise than the floating point granularity of the machine. When 4-byte IEEE floating point math is used, the smallest possible fractional error in any computation will be about $1\text{E-}7$. For 8-byte IEEE math, the smallest possible fractional error will be about $1\text{E-}15$. For the full 10-byte IEEE math used in 80x87s, the smallest attainable fractional error is in the vicinity of $1\text{E-}19$.

Given that TableCurve 2D uses the full hardware precision 10-byte floating point for its standard linear fitting, you might think that finite floating point precision issues would not arise. This is certainly true for the linear equations with relatively few terms. It is not true, however, for higher order polynomial and rational equations.

Again, let us consider the following TableCurve 2D polynomial:

$$y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10}$$

The design matrix for this equation will have a certain condition number. This number approaches infinity as the matrix becomes nearly singular. For the 64 bit mantissa of 10-byte IEEE floating point, a condition number greater than 264 ($1.8\text{E}19$) means significant roundoff and truncation losses will occur in conventional matrix processing.

As a matrix becomes more ill-conditioned or closer to singular, its condition number increases toward infinity and the determinant decreases toward zero. Such ill-conditioning plays havoc with conventional matrix procedures such as Gaussian Elimination, Gauss-Jordan, and LU-Decomposition.

As a rule of thumb, the maximum order polynomial that can be fitted conventionally is about tenth order for long double 10-byte math, and about eighth order for double precision 8-byte math. In general, all accuracy with a conventional algorithm can be considered lost when the condition number matches the floating point granularity ($1\text{E}+19$). This does not mean that a fit will look poor, however.

If data were generated for a tenth order polynomial to a full 19-digit precision using a set of properly scaled coefficients, it is reasonable to expect that the fitting algorithm should recover these values to some level of precision. At some stage in increasing order, however, all accuracy is lost in the recovery of these coefficients via a conventional algorithm.

SVD

Singular Value Decomposition, SVD, is a matrix procedure designed to handle ill-conditioning. The SVD procedure produces a vector of singular values. When the range of these singular values exceeds some optimum range based upon the floating point precision of the machine, the smallest singular values are set to zero.

This actually translates to the discarding of matrix information. The hope is that the information is so corrupted with error, that its omission is less harmful than its inclusion. In the case of the kinds of matrices generated by TableCurve 2D, this is not certain. In some instances, SVD will produce a far more stable set of coefficients with very nearly the same goodness of fit. In other cases, this zeroing of information may result in useless coefficients and a drastically poorer goodness of fit.

As a matrix becomes singular, the accumulated roundoff and truncation error in conventional matrix procedures tends to produce delicately balanced offsetting coefficients that push toward positive and negative infinity. Such coefficients are said to be unstable because a very small change in even a single input data point might result in a very different set of coefficients. Such fits are considered unstable

because they can rarely be extrapolated in any way.

When a nearly singular design matrix is successfully processed by SVD, the fitted coefficients will be more stable. They will actually be forced toward zero rather than infinity by the floating point limitations SVD successfully manages. As a result, SVD fitted coefficients will look nothing like those fitted by conventional methods. They will be smaller in magnitude, and they will span a much smaller dynamic range. Small changes in input data will still result in similar coefficient values, and the fitted curve can often be extrapolated in a limited way with some success.

Limitations of SVD

If SVD is such a useful procedure, why does TableCurve 2D not use it by default for all linear fitting? One reason is computation time since the SVD procedure is about an order of magnitude slower than conventional matrix solution methods. Another reason is the unpredictable nature associated with this zeroing of singular values. Sometimes a significantly poorer least-squares fit results. Yet another reason is that for simpler equations where ill-conditioning is not an issue, the conventional methods produce a higher precision.

Limitations of Least Squares

In the one-step matrix solution for the previous example, the matrix contained an x^2 element even though the equation contained no more than an x basis function. Minimizing the square of the residuals rather than a more linear measure, such as absolute value of the difference, affects more than just computational precision.

Consider the example of two points that have exactly the same x value. Assume each point also has the same weight or standard deviation, meaning each is weighted equally in computing the sum of squared residuals. If one of these points is twice the y -distance from the curve as the other, its influence in this sum of squared residuals will be four times greater than that of the closest point.

Outliers

This raises two important limitations to be aware of in least squares fitting. The first is that outliers can significantly influence the fitted surface and its coefficients.

By Standard Error

TableCurve 2D offers a variety of ways to recognize such outliers. When points are colored by their relationship to the overall standard error of fit, it is easy to spot points that lay outside 3 standard errors. In the default colors, such points will be red. Such points can be turned off and the data refitted without them. You may also wish to consider the merit of disabling points that lay outside 2 standard errors of the fit. These points have a yellow color by default and represent points which lay outside 95.4% of the error distribution for the fit.

You must keep in mind that this coloring by standard error is based on the overall least squares solution. This does not take into account the magnitude of the y -value nor how accurately a curve may be determined in any particularly region.

For example, consider a transition curve that begins at $y=0$ for the initial x values and which rises to $y=100$ for the final x values. Let us assume the standard error for this curve fit is 3.0 and that the curve is very accurately determined at the lower and upper plateaus, but very poorly in what is a very sharp transition.

All points will be colored based upon this 3.0 standard deviation for the fitted line. The accurately determined points with a y value near zero will be treated in exactly the same way as those accurately determined at the $y=100$ upper plateau and these will be treated exactly as those marginally determined points with a y -value near 50 in the middle of the transition. If the observed y value for a point is outside this ± 9 , it is colored red and said to be outside 3 standard deviations for the fit. If it is between ± 6 and ± 9 , it is colored yellow, regardless of the magnitude of the y -value, nor how accurately the curve is determined at the x of concern.

Perhaps a point should genuinely be an outlier on the lower or upper plateau if its y -value is ± 1 of the computed value. Such points do not appear as outliers because of the large impact of the points inaccurately determined in the transition.

By Confidence Intervals

The confidence intervals for a curve measure, at a specific x , the y range associated with a specified probability for the true y value. It is used most often when data consists of averages and inverse variance weights, or when data consists of multiple appended sets. Prediction intervals for a curve are likewise measured at a given x , and consist of the y range associated with a specified probability for the y value of the next experiment. Prediction intervals are commonly used when data consists of a single experiment,

and you are interested in predicting the values for the next experiment.

Prediction intervals are also a useful method for discerning and disabling outliers in the data. Since an interval is a more local measure of confidence, whether or not a point lay outside a given probability also depends on how effectively the curve may be characterized by the curve-fit in that data region. As a rule, a curve-fit tends to be best determined near the mean of the x values and less so near the edges of the data region.

Whether outliers are removed by standard error or by prediction interval seems largely a matter of taste. In most cases, points outside a 95% prediction interval will also lay outside two standard errors of fit.

By Error Bars

If the data has been constructed from replicate measurements, either by external compilation or from within TableCurve 2D, errors bars will be available. These can be used to spot individual measurements which may be in question and should not be included in the composite y values.

Y Data Spanning Orders of Magnitude

The other major limitation in least squares occurs when the y-data spans many orders of magnitude. Imagine a data set whose y axis is on a log scale ranging from $1E-5$ to $1E+5$. An error of $1E-6$ may be typical at $1E-5$ values whereas an error of $1E+4$ might be typical at $1E+5$. The squared residual for the typical error at $y=1E-5$ would be $1E-12$ whereas the squared residual for a typical error at $y=1E+5$ would be $1E+8$. When $1E+8$ is added to $1E-12$, even with Intel 80 bit hardware floating point, the result is still $1E+8$. The error at the low y-value is insignificant in the overall sum of squared residuals. As such, the curve fit is unlikely to fit the data at these low y-values.

Weighting for Magnitude of Errors

When errors vary as a constant fraction of the y-value, one solution is to weight the data for the magnitude of y. In TableCurve 2D, the simple calculation $W=1/(Y*0.05)^2$ would adjust for y magnitude when the errors are approximately 5% of the y magnitude. This type of weighting scheme is rather dependent on having a constant fractional error.

Another solution is to simply plot and fit data that has been pre-transformed to $\ln(y)$. In TableCurve 2D, all statistics and all fitting are based upon y. Even when an equation is y-transformed, the goodness of fit statistics are computed based upon y rather than upon the scale of the transform. TableCurve 2D must assume that the y variable represents the world scale of the data. For equations to be comparable, this means that all data must be based upon a common y-measure of error. One equation cannot report statistics based on minimizing the $\ln(y)$, another on minimizing the inverse of y, and yet another on minimizing y itself. All measures of a curve fit must be scaled to the dimensions of the original y-variable.

Log(Y) or Ln(Y) Pre-Transform

This is where you can help yourself free of the limitations of the least squares process. A simple $Y=\text{LOG}(Y)$ calculation will convert the data from a Y scale to a base 10 logarithmic scale. The $1E-5$ to $1E+5$ data that spanned ten orders of magnitude now spans from -5 to +5. Weighting suddenly becomes less critical since the errors at all data points will significantly influence the overall least squares minimization.

The problem is solved by changing the scale of the problem from one that plays havoc with least squares and the limitations of floating point hardware to one that utilizes nearly all of the precision in the machine.

1/Y Inverse Transform

Often times an inverse transformation will work quite effectively in changing the scale of a problem. For data beginning at $y=1$ and increasing to very large y values, a calculation $Y=1/Y$ will result in all Y data being between 0 and 1. This is the optimum dynamic range for floating point processing.

As a general rule, if your data spans only a few orders of magnitude, you should find TableCurve 2D's y -based least squares minimization to be fully satisfactory. Once your data starts to span more orders of magnitude, you should seriously consider assisting TableCurve 2D by redefining the scale of your problem. You may also find that by minimizing the squared residuals in $\ln(Y)$ or $1/Y$ rather than Y itself, you find a host of additional linear equations that you would not have otherwise realized.

For non-linear equations, and for those linear equations you might be willing to fit via a **UDF**, there is another especially attractive alternative. By using the **robust minimizations** available in TableCurve 2D's non-linear fitting engine, it is possible to deal with outliers and Y -dynamic range in a very effective and automatic fashion. It should be noted that **non-linear robust fitting** represents a very non-traditional approach to curve-fitting because least squares is altogether abandoned in favor of minimizations which are far less sensitive to outliers and Y -dynamic range.

Robust Fitting



By selecting a robust minimization procedure in the **Curve Fit Preferences** option or within the separate customization dialogs for the **Curve-Fit Peak Functions**, the **Curve-Fit Transition Functions**, or the **Curve-Fit Kinetic Equations** option, robust fitting can be done for any of TableCurve 2D's non-linear equations, including **UDFs** and **TCEX.DLL functions**.



The Process menu also includes a **Curve-Fit Robust Straight Line** option which fits the simple straight line by least squares and by all three of TableCurve 2D's robust minimizations.

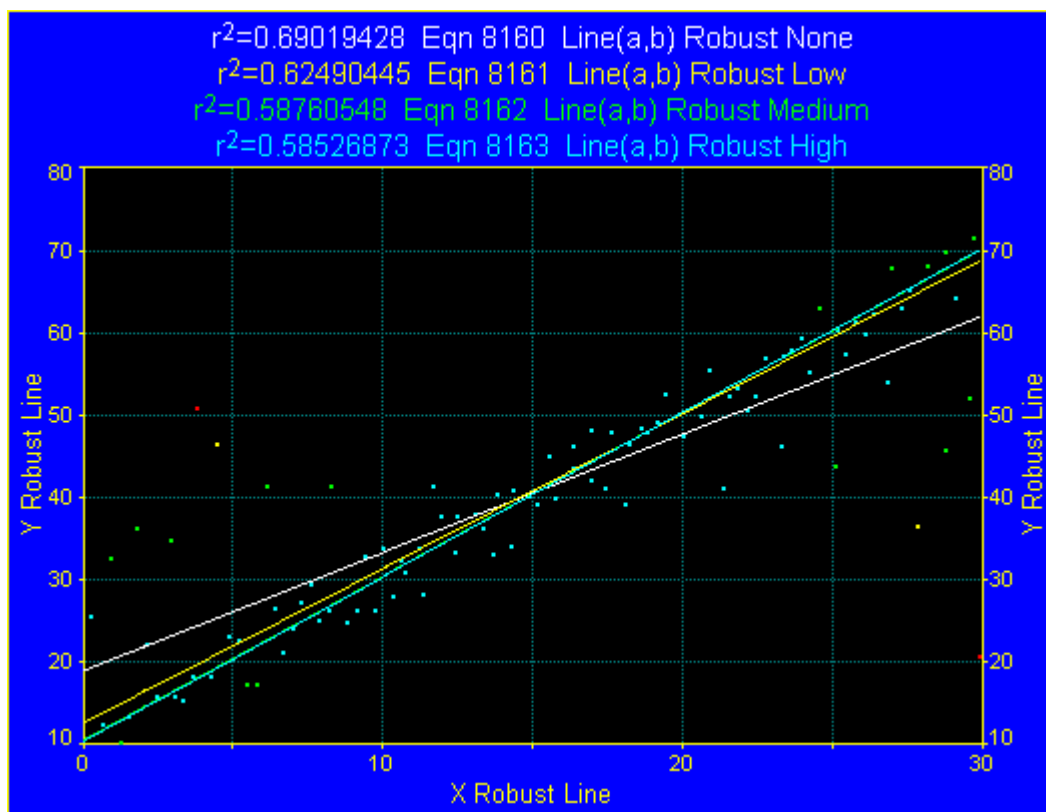
Limitations of Least-Squares

Least-squares fitting involves the minimization of the sum of the squared residuals. There are two instances where this minimization produces a less than satisfactory fit. The first is where significant outliers are present. In this case, the square of the residuals of these outlier points may, within a given region, significantly shift the fitted curve away from the bulk of the data. The other instance is when the Y-data spans more than several orders of magnitude. The squared residuals of the largest valued Y-points can overwhelm the influence of the squared residuals of the smallest Y-valued points, causing the smallest Y-value points to either be poorly fitted or not fitted at all. Data that requires a logarithmic Y-scale to see all of the points may be a good candidate for robust fitting, especially if four or more major log divisions are present.

Managing Outliers

The Gaussian or normal distribution is a very compact one. The points that lay outside ± 3 SD, which by default are red, should statistically occur only once for every 370 points (99.73%). If you are one of the TableCurve 2D users who sees a much higher frequency of points outside 3 standard deviations, one solution is to simply **exclude the outliers and refit**. The exclusion will be based on the residuals for a specific equation, so you will want to select your desired equation before isolating the outliers, excluding them, and refitting.

In some instances, this removal of outliers isn't straightforward. This exclusion based on ± 3 SD is based upon a least squares assumption of normally-distributed or Gaussian errors. Life isn't so simple when errors are far from Gaussian or unevenly distributed. Consider the following noisy straight line data set:



The least-squares straight-line curve-fit (white curve) clearly reveals the impact of outliers which significantly shift the fitted curve away from the bulk of the data. In this data set there are only two points outside 3 standard deviations and similarly only two between 2 and 3 standard deviations. When these four points are excluded and the data is refitted, the curve shifts somewhat closer to the bulk of the data.

Still, there is a sense that it is "likely" the curve should shift a great deal more toward the bulk of the data. What we should like then is a method that automatically offers a more robust fit, one that is far less sensitive to data points which are unlikely to be valid. With least squares, points less likely to be valid factor into the fit far more than those likely to be more accurate. TableCurve 2D's non-linear engine offers exactly this type of robust or maximum likelihood fitting.

Three Optional Robust Procedures

TableCurve 2D's non-linear fitting offers three robust methods that can be used instead of least squares. These methods are built directly into the non-linear engine enabling all non-linear functions, including UDFs and external C/Fortran functions, to be fitted by a robust method. The following graphs reflect the improvement realized by these three robust methods. In these examples, no points have been excluded from the fitting.

The Robust Low curve-fit (yellow reference) shows the improvement realized by least absolute deviation fitting. The Robust Medium curve-fit (green reference) is for the Lorentzian minimization, and the Robust High (blue reference) is for a robust procedure using the limiting form of the Pearson VII function. All three methods are more effective in offering the most likely fit, that which includes the bulk of the data. In the latter two, note that outliers are essentially ignored as they minimally influence the fit.



By selecting a robust minimization procedure in the [Curve Fit Preferences](#) option or within the separate customization dialogs for the [Curve-Fit Peak Functions](#), the [Curve-Fit Transition Functions](#), or the [Curve-Fit Kinetic Equations](#) option, robust fitting can be done for any of TableCurve 2D's non-linear

equations, including UDFs.



The Process menu also includes a **Curve-Fit Robust Straight Line** option which fits the simple straight line by least squares and by all three of TableCurve 2D's robust minimizations.

When to Use Robust Methods

There are two instances where a robust method is particularly attractive. The first is the case shown in the above example where significant outliers are present and whose presence is adversely affecting the fitted curve. The other instance is when the y-data spans more than several orders of magnitude. As mentioned in the previous section, the squared residuals of the largest valued y-points can overwhelm the influence of the squared residuals of the smallest y-valued points, causing the smallest y-value points to either be poorly fitted or not fitted at all. Data that requires a logarithmic Y-scale to see all of the points may be a good candidate for robust fitting, especially if four or more major log divisions are present.

Least Absolute Deviation

The essence of robust fitting is to use a minimization that is less influenced by outliers and the dynamic range of the Y-variable. Instead of minimizing the sum of the squares of the residuals, an obvious alternative is to minimize the sum of the absolute value of the residuals:

$$\sum |y_i - \hat{y}_i|$$

This is probably the best known robust method, though not necessarily the best, and is usually designated as least absolute deviation. Of TableCurve 2D's three robust methods, least absolute deviation is the least powerful in terms of managing outliers.

Lorentzian Minimization

The intermediate method in terms of power of robustness is a Lorentzian minimization. Here what is actually being minimized is:

$$\sum \ln(1 + |y_i - \hat{y}_i|^2)$$

If you are uncertain of which robust method to use, we strongly recommend this Lorentzian minimization. It is very effective when you have noisy data with outliers or if your data spans many orders of magnitude in Y. These fits also tend to converge as rapidly as the least absolute deviation minimization.

Pearson Minimization

This is the most robust of the three methods. Here the minimization is:

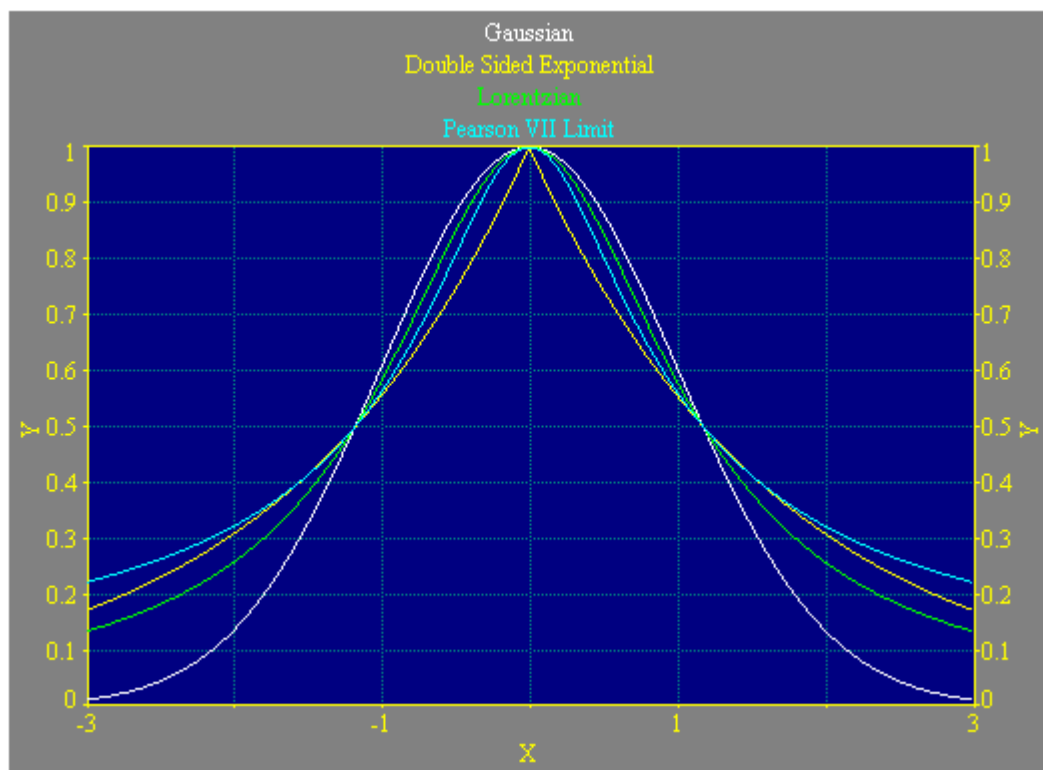
$$\sum \ln\left(\sqrt{1 + |y_i - \hat{y}_i|^2}\right)$$

With this method, outliers tend to have almost no impact on the fitted curve. This minimization represents an extreme one where wild and random errors are expected as a natural course.

Gaussian Error Distribution

Each of the various minimization formulas corresponds with a maximum likelihood probability distribution. For least squares, this corresponding error distribution is Gaussian or normal. Relative to fit standard error (SE), a least squares goodness of fit, 95.4% of the data points should have residuals with a magnitude less than 2 SE (1 in 21.98 points should lay outside 2 SE). The value is 99.73% for residuals within 3 SE (1 in 370.4 points should lay outside 3 SE). The Gaussian distribution decays very rapidly. It suggests that only 1 of 15787 points should lay outside 4 SE, only 1 of 1.74 million points should lay outside of 5 SE, and those outside 6 SE should be only 1 in a half billion. In the real world where data is subject to human error, it is generally agreed that major errors are more likely or probable than the Gaussian distribution suggests.

The following graph plots the error distributions. These are normalized so that amplitudes and FWHM (half-maximum widths) are equal.



Double Sided Exponential Error Distribution

The least absolute deviation fitting corresponds with a double-sided exponential probability distribution. While such a distribution produces reasonably wide tails, it also has a discontinuous first derivative at the peak center. As such, the actual error profile isn't likely to match this double-sided exponential in this center region. The tails of the double-sided exponential may however, be very practical.

This distribution suggests that 75.7% of the points should be within 2 SE and 88.0% within 3 SE. This distribution expects 1 of 16.9 points to be outside 4 SE, 1 of 34.3 to be outside 5 SE, and 1 of 69.6 to be outside of 6 SE. It is important here to recognize that such SE-relative values are based upon a least squares goodness of fit standard deviation.

Lorentzian Error Distribution

The Lorentzian minimization is strongly recommended for data with significant outliers because the Lorentzian distribution is a very natural one both at the center and the very wide tails. Such broad tails in

effect state that significant errors are expected or likely and that points with such errors should minimally influence the fit.

The Lorentzian distribution suggests that 68.5% of the points should be within 2 SE and 83.3% within 3 SE. In terms of outliers, the Lorentzian expects 1 of 7.9 points to be outside of 4 SE, 1 of 9.8 to be outside of 5 SE, and 1 of 11.4 to be outside of 6 SE. Even out to 10 SE, the Lorentzian contains only 94.9% of the points, less than the Gaussian contains within 2 SE.

Pearson VII Limit Distribution

TableCurve 2D also offers a distribution with the same naturalness about the peak center as the Lorentzian but with extremely wide tails. This is based upon the Pearson VII function with a power term of 0.5. This is the smallest power term that can be used in the Pearson VII function and still have the peak converge to a finite area.

It is difficult to estimate areas for this distribution since area convergence does not occur until near infinity. If you know your model is very appropriate to your data, but that there is a high likelihood that a significant number of the data points are very seriously in error or perhaps even fabricated, this highly robust method may be a good choice for removing the impact of such points.

Analysis of Residuals

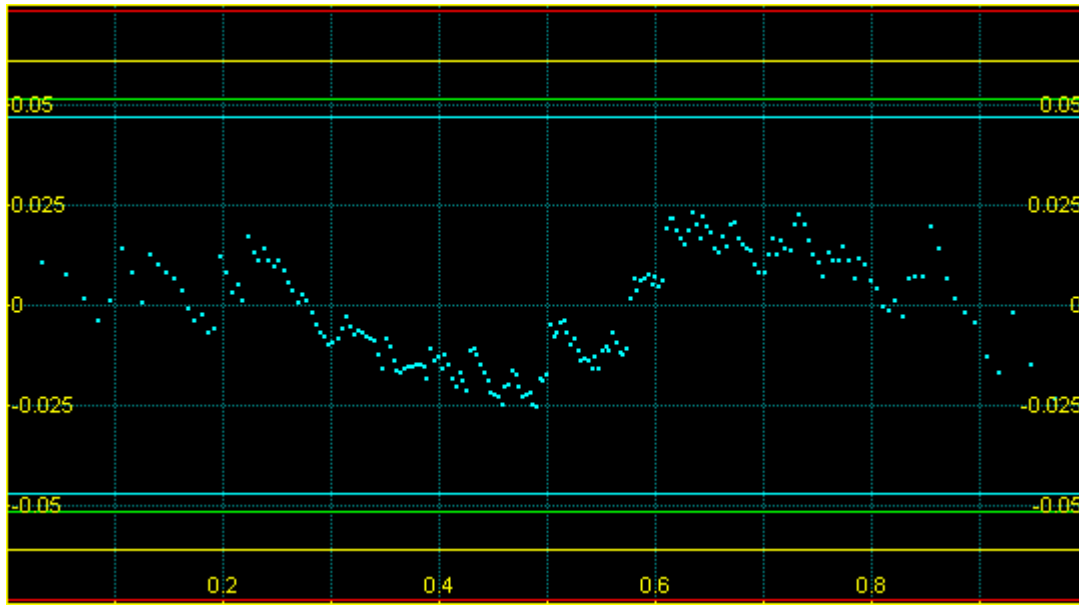


The [Display Residual Distribution](#) format for the [Residuals Graph](#) displays a histogram of the residuals. Distributions with obvious asymmetry or wide tails would readily disqualify this assumption of Gaussian errors.



The [Display Residuals in Stabilized Normal Probability Plot](#) format for the [Residuals Graph](#) is the preferred approach in TableCurve 2D for assuring errors are normal. A stabilized normal probability (SNP) plot uses an arcsine transformation on both X and Y to produce a normal probability plot that uses a linear scale for both the X and Y axes. On such a plot, perfectly normal errors plot as a 45 degree line. Critical limits also have a 45 degree slope, and lay equally above and below this line. TableCurve 2D modifies the SNP slightly and uses a delta SNP, where the X value is subtracted from the Y. This produces a horizontal $y=0$ for pure normal data, and horizontal critical limit lines.

TableCurve 2D plots 90, 95, 99, and 99.9% critical limit lines on the SNP plot. A 99% critical limit means that in only 1 out of 100 data sets should even a single point violate this limit. You may find the 99% critical limit the most useful. If even a single data point in the SNP violates this 99% limit, it is reasonable to assume that the errors fail this normality test. The following graph is from a curve fit which yielded normal errors.



By default, the 90% critical limit lines will be blue, the 95% green, the 99% yellow, and the 99.9% red. You should inspect the SNP before attempting to use the parameter confidence statistics or confidence or prediction intervals in any way.



The **Error Models** option in the **Residuals Graph** performs non-linear least-squares fits for the four error models and furnishes an r^2 value for each.

You should not assume simply because a distribution of errors is fitted well by a Gaussian that a robust procedure is if no value. Two key reasons for using a robust minimization are to deal effectively with outliers and a wide dynamic range on the Y-variable. It is very possible that such conditions will not be reflected in the fit of the overall residuals distribution.

Evenness of Residuals Distribution

In the noisy line example, the residuals are not evenly distributed across the X-range of the data. The errors at one end or in one region of the X-range have one sign and those at the other end or in another region have the opposite sign. In such cases, a more robust method may be desirable to lessen the impact of this uneven distribution of residuals.

Using Least-Squares

If the Y-range spans no more than a few orders of magnitude, there are no obvious outliers, and the errors appear to be normally distributed, you should use least-squares minimization. Most distributions in nature tend toward the Gaussian, and least-squares is perfectly sufficient most of the time. You should bear in mind that the robust methods do not have the same dynamic for convergence, and additional iterations and lengthier fits will be the price paid for utilizing one of the robust methods. Note that it may be necessary to set the maximum iterations to a higher count to insure convergence.

Non-Linear Nature of Robust Minimization

Whereas a non-linear iterative procedure is required in least-squares fitting whenever coefficients appear non-linearly in the equation, a non-linear procedure is always required for robust fitting. This means that the robust fitting of even the simple straight line equation must be done non-linearly. The **Curve-Fit Robust Straight Line** option in the Process menu uses the non-linear fitting engine with all four

minimizations to fit the simple line equation. Although it is not necessary to fit the least squares minimization non-linearly, this is done simply to be consistent with the three robust minimizations which do require this non-linear fit.

Fitting a Linear Equation by a Robust Procedure

To fit any linear equation by a robust procedure, it must be fit non-linearly. For this reason there is now a **Save TableCurve UDF** option in the Review which will create a UDF for any of TableCurve 2D's linear equations having ten or fewer terms, excepting the Chebyshev and Fourier series models. Such a generated UDF can then be fitted with a robust method as specified in the **Curve Fit Preferences** option.

Curve-Fit Statistics

To maintain a true basis for comparing the linear and non-linear equations in TableCurve 2D, all **goodness of fit statistics** are based upon sum-of-squares, even when a fit is made using a robust method. The r^2 , DOF adjusted r^2 , Fit Standard Error, and F-statistic are all based upon the sum of squared residuals. Even though a robust fit does not involve a least squares minimization, a sum of squared residuals is still computed in order to generate these numeric indices of fit.

The computation of standard errors and confidence ranges, as well as prediction and confidence intervals, are also based upon sum-of-squares computations and can be assumed accurate only when the residuals are normally distributed. If a chosen critical limit in the **SNP** is violated, or if the residuals distribution has narrower or wider tails than the Gaussian, or has appreciable asymmetry, then this assumption of normality fails, and these statistics should not be used as absolute measures of uncertainty. This is true for both least-squares and robust minimizations.

Graphical Inspection

Because these **goodness of fit statistics** for a robust minimization use a least squares measure, and no least squares optimization occurs, it is certain that these goodness of fit values will be inferior to those produced for the actual least squares fit. In many cases, the statistics for a robust fit will approach that of the least squares fit but they will never equal it. Thus in the option that fits a straight line by all four minimizations, the least squares case will always be at the top of the list, although any or all of the robust fits may be superior in being less influenced by outliers or in managing a wide dynamic Y range. This is simply one more reason to rely on graphical inspection as to the value of a robust fit. The instances where a robust fit is less influenced by outliers or where it fully manages a wide dynamic range in Y will not be apparent from the numeric indices of fit. Graphical inspection is essential.

Numerical Methods

Linear Fitting

All TableCurve 2D linear fitting algorithms except the high precision polynomials and rationals use 80 bit floating point math. This is the full 19-digit precision of Intel floating point units. This highest precision was chosen because most matrix operations are highly susceptible to precision losses and roundoff errors. Using software other than TableCurve 2D, even with the same procedures, you may find different fitted coefficient values for a given equation if such programs use 32 or 64 bit precision.

All of TableCurve 2D's linearly fitted equation use a single step matrix solution method for the coefficients and a single step matrix inversion for the standard errors and confidence intervals.

The default (Fast Std) linear fitting procedures are as follows:

Equations	Coefficients	Errors/Confidence Intvls
1-2000	Direct Matrix Solution	LU Decomposition
2001-8000 exc. high prec	Gaussian Elimination	LU Decomposition
High Precision Poly/Ratl	Gaussian Elimination*	Gauss-Jordan*

*The high precision polynomials and rationals use a 38-digit precision math emulator and special matrix routines which implement this higher precision floating point.

Optionally, for all except the high precision fits, a single method can be selected for resolving the coefficients, standard errors, and confidence intervals. The options include the Gauss-Jordan, LU-Decomposition (LU), and Singular Value Decomposition (SVD) procedures.

In a simple equation, the direct matrix, Gaussian Elimination, Gauss-Jordan, LU Decomposition, and Singular Value Decomposition methods are likely to produce identical coefficient values. As such, TableCurve 2D's default uses the fastest methods for its simpler equations.

A separate set of built-in procedures, Fast SVD, is available where the SVD algorithm is used for fitting the standard precision 6000 series polynomial equations and the 7000 series rational equations. Higher order polynomials and rationals tend to have an ill-conditioned design matrix, exactly the type of situation where the SVD procedure can sometimes produce more stable results.

The SVD procedure manages ill-conditioning by pushing the coefficients closer to zero rather than the delicate balance offered by the other methods where the coefficients are driven toward very large positive and negative offsetting values. With higher order polynomials and rationals, the SVD procedure often produces unexpected though more stable coefficient values. The price for the added stability the SVD method offers is a reduced goodness of fit. Also, SVD requires almost an order of magnitude more fitting time than the other methods.

For more information on linear fitting algorithms, the following references may be of interest:

- Bevington, Philip R. 1969, Data Reduction and Error Analysis for the Physical Sciences (New York: McGraw Hill)
- Dongara, J.J. et al. 1979, LINPACK User's Guide (Philadelphia: Society for Industrial and Applied Mathematics)
- Press, William H. et al. 1992, Numerical Recipes in C; The Art of Scientific Computing (Cambridge,UK: Cambridge University Press)

Non-Linear Fitting

Non-linear equations cannot be solved in a single step solution of a matrix, but must be managed in an iterative fashion. As such, non-linear fitting is a much slower process and one that often requires starting estimates in order to initiate the fitting.

TableCurve 2D's non-linear fitting engine is a 80-bit Levenburg- Marquardt algorithm that uses the Gauss-Jordan procedure for the matrix inverse required in each iteration. In non-linear fitting, the parameters are iteratively adjusted to minimize a goodness of fit merit function. If the algorithm is fully successful, a true global minimum is achieved.

When fitting any equation using a robust minimization, a non-linear iterative method is required even if the equation is a linear one. TableCurve 2D employs its own custom modifications of the Levenburg-Marquardt algorithm to enable three powerful robust minimizations in addition to least squares. These robust methods are explained in Chapter 11.

Since the Levenburg-Marquardt method's minimization procedure requires the partial derivatives with respect to the parameters, TableCurve 2D uses analytic derivatives for the built-in non-linear functions for the highest precision and the most rapid fitting and convergence.

The Levenburg-Marquardt algorithm also requires starting estimates for the adjustable parameters. For the built-in non-linear equations, these are automatically supplied using pre-scan values specific to the current data profile.

The non-linear fitting algorithm can be configured for the maximum number of iterations as well as to specify a convergence criteria. TableCurve 2D deems the algorithm converged when the r2 coefficient of determination is unchanging in the significant digit specified for five consecutive iterations. The fitting engine can also be interrupted to abort a non-linear fit that is going nowhere, but is still slowly changing in the r2 value.

TableCurve 2D's fifteen user-defined functions are a part of the non-linear fitting engine. Here the starting estimates must be supplied by the user. User-functions are compiled and fitted in 80 bit precision, using numeric derivatives. Successful convergence may well depend upon the quality of the starting estimates.

Similarly, the up to 100 **TCEX.DLL FORTRAN/C external functions** are fitted using access points into the non-linear fitting engine.

For more information on the Levenburg-Marquardt non-linear fitting algorithm, see the Bevington and Press references previously given.

Integration, Minimization, And Root-Finding

TableCurve 2D uses the Gaussian Quadrature method for the main integration of the fitted equations and for cumulative areas. The Quadrature procedure's convergence is targeted for eight significant figures. In the Evaluation option, the precision actually achieved is reported.

In the AI() and AIP() integration functions used in UDFs, a Gaussian Quadrature procedure is first used. If the target precision is not achieved, a Romberg procedure is then attempted. If neither algorithm achieves the target precision, the UDF will use the value from the procedure that came closest.

Brent's method is used for determining the minima and maxima of the fitted equation and its derivatives. A partition approach is used to limit the possibility of reporting local minima.

The root-finding routine in TableCurve 2D and in its generated code is a multi-partition bisection procedure.

The roots of rational denominators are found using an eigenvalue procedure.

For more information on these methods, please refer to the Bevington and Press references.

Statistical

Sum of Squares due to Error	$SSE = \sum_{i=1}^n w_i (y_i - \hat{y}_i)^2$ $w_i = \text{weight}$ $y_i = y \text{ data value}$ $\hat{y}_i = \text{predicted } y \text{ value}$ $n = \text{number of data points}$

Sum of Squares about Mean	$SSM = \sum_{i=1}^n w_i (y_i - \bar{y})^2$ $\bar{y} = \text{mean of } y \text{ values}$

Coefficient of Determination	$r^2 = 1 - \frac{SSE}{SSM}$

Degree of Freedom	$DOF = n - m$ $m = \text{number of coefficients fitted}$

Degree of Freedom Adjusted r^2	$DOF \text{ Adj } r^2 = 1 - \frac{SSE(n-1)}{SSM(DOF-1)}$

Mean Square Error	$MSE = \frac{SSE}{DOF}$

Fit Standard Error (Root MSE)	$S_e = \sqrt{MSE}$

Mean Square Regression	$MSR = \frac{SSM - SSE}{m - 1}$

F-statistic	$F = \frac{MSR}{MSE} = \frac{\frac{SSM - SSE}{m - 1}}{\frac{SSE}{DOF}}$

Confidence Interval $\hat{y} \pm t\sqrt{MSE} \sqrt{l'(X'X)^{-1}l}$
 t = Student's t for confidence level and DOF
 l = coefficient partial derivative vector evaluated at x_i

Prediction Interval $\hat{y} \pm t\sqrt{MSE} \sqrt{1 + l'(X'X)^{-1}l}$
 $(X'X)^{-1}$ = inverse of design matrix

Stabilized Normal
Probability X $SNP_{x_i} = \frac{2}{\pi} \arcsin \left(\sqrt{\frac{i - 1/2}{n}} \right)$

Stabilized Normal
Probability Y $SNP_{y_i} = \frac{2}{\pi} \arcsin \left(\sqrt{\frac{1}{2} \operatorname{erf} \left(\frac{(y_i - \hat{y}_i) - R_m}{\sqrt{2}R_y} \right) + \frac{1}{2}} \right) - SNP_{x_i}$
 R_m = mean of residuals
 R_y = $n - 0$ based standard deviation of residuals

Y-Transformed Equations

Conversions For Y-Transformed Equations

TableCurve 2D offers 530 Y-transformed equations. A transformed equation is expressed as $f(y)=...$ where $f(y)$ is one of the following:

$\ln(y)$
 $1/y$
 $\sqrt{y}$
 y^2

Y-transformations enable non-linear equations to be fitted by linear procedures.

Special Weighting of Y-Transformed Equations

TableCurve 2D applies a secondary weighting to compensate in the fitting process for these y-transformations. This weighting is fully transparent to the user. You may, however, observe some unexpected confidence and prediction intervals as a result of this secondary weighting.

The following list summarizes the TableCurve 2D Y-transformed equations which have familiar non-linear forms.

Non-Linear Form	TableCurve 2D	Conversion
$y = a'x^{b'}$	$34 \ln y = a + b \ln x$	$a' = \exp(a), b' = b$
$y = a' \exp(b'x)$	$22 \ln y = a + bx$	$a' = \exp(a), b' = b$
$y = \frac{x}{a'x + b'}$	$59 \frac{1}{y} = a + \frac{b}{x}$	$a' = a, b' = b$
$y = a'b'^x$	$22 \ln y = a + bx$	$a' = \exp(a), b' = \exp(b)$
$y = a'x^{b'x}$	$23 \ln y = a + bx \ln x$	$a' = \exp(a), b' = b$
$y = a'b'^{1/x}$	$38 \ln y = a + \frac{b}{x}$	$a' = \exp(a), b' = \exp(b)$
$y = a'x^{b'/x}$	$37 \ln y = a + b \ln x/x$	$a' = \exp(a), b' = b$
$y = a' \exp(b'/x)$	$38 \ln y = a + \frac{b}{x}$	$a' = \exp(a), b' = b$
$y = \frac{1}{a' + b'x}$	$43 \frac{1}{y} = a + bx$	$a' = a, b' = b$
$y = \frac{1}{a' + b' \ln x}$	$55 \frac{1}{y} = a + b \ln x$	$a' = a, b' = b$
$y = a'b'^x x^{c'}$	$1222 \ln y = a + bx + c \ln x$	$a' = \exp(a), b' = \exp(b), c' = c$
$y = a'b'^{1/x} x^{c'}$	$1388 \ln y = a + b \ln x + \frac{c}{x}$	$a' = \exp(a), b' = b, c' = \exp(c)$

Non-Linear Form	TableCurve 2D	Conversion
$y = a' \exp \left[-\frac{1}{2} \left(\frac{x - b'}{c'} \right)^2 \right]$ Gaussian	1213 $\ln y = a + bx + cx^2$	$a' = \exp \left(a - \frac{b^2}{4c} \right)$ $b' = -\frac{b}{2c}$ $c' = \sqrt{\frac{-1}{2c}}$
$y = \frac{a'}{1 + \left(\frac{x - b'}{c'} \right)^2}$ Lorentzian	1423 $1/y = a + bx + cx^2$	$a' = \frac{c}{ac - b^2/4}$ $b' = \frac{-b}{2c}$ $c' = \frac{\sqrt{ac - b^2/4}}{c}$
$y = a' \exp \left[-\frac{1}{2} \left(\frac{\ln(x/b')}{c'} \right)^2 \right]$ Log-Normal	1357 $\ln y = a + b(\ln x)^2 + c \ln x$	$a' = \exp \left(a - \frac{c^2}{4b} \right)$ $b' = \frac{-c}{2b}$ $c' = \sqrt{\frac{-1}{2b}}$

Other Transformed Equations

If you wish to see if a two-parameter or three-parameter non-linear equation has a counterpart within TableCurve 2D's Y-transformed equations, simply enter the non-linear equation as a calculation, apply it to the data table, and fit the data using only the Y-transformed equations to see if any equation has a perfect r^2 of 1.0.

For example, entering the calculation $Y=10*X^{1.5}$, applying it to a data table, and fitting it only to the Y-transformed equations results in Equation 34 with an r^2 of 1.0 at the top of the list. The fitted b coefficient is 1.5 and the fitted a is the $\ln(10)$.

Limitations Of Y-transformed Equations

The Y-transformations make it possible to fit certain non-linear equations by a linear method. The fitting is thus accomplished in a fast single step matrix solution without initial estimates. Since TableCurve 2D automatically **weights** the data to compensate for the effects of the transform, it is possible to achieve effective fits which are nearly as good as true iterative non-linear fits.

Equations linearized by Y-transforms are not, however, the original equations. With any degree of noise, a more accurate fit will be achieved using a true non-linear iterative procedure. If the linearized form involves the log or square root transform, no fit will even occur if there are Y-values present less than

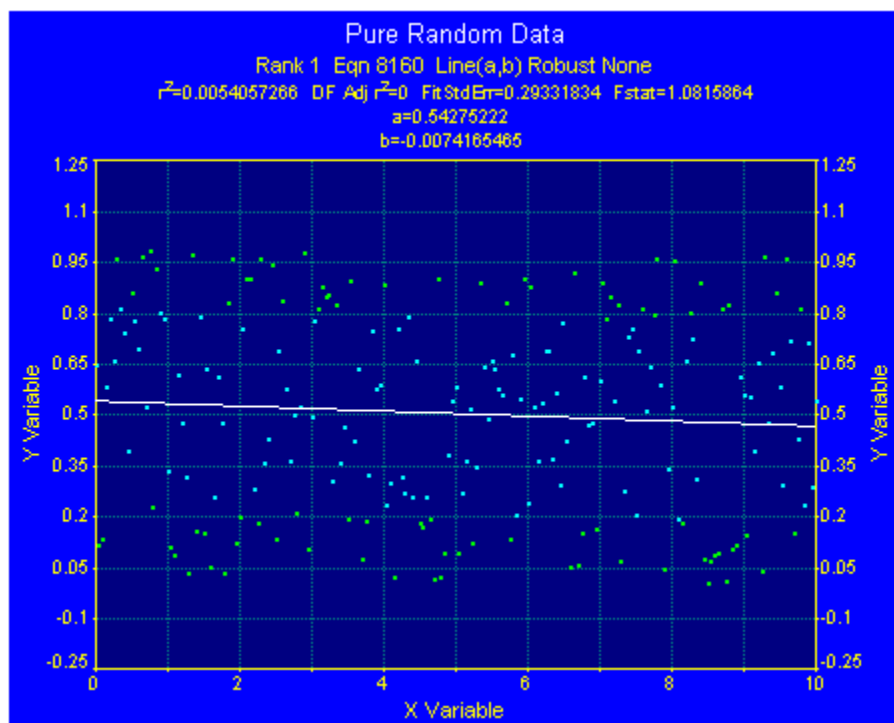
zero. In such cases, you will have to use the [X-Y sectioning](#) or the [TableCurve Editor](#) to exclude all points with Y-values less than zero.

Curve-Fitting Pitfalls

The purpose of this section is to illustrate some of the pitfalls that exist in curve-fitting science, and some of the ways they can be avoided. TableCurve 2D contains a significant number of features designed specifically to prevent the kind of errors that are common to curve-fitting.

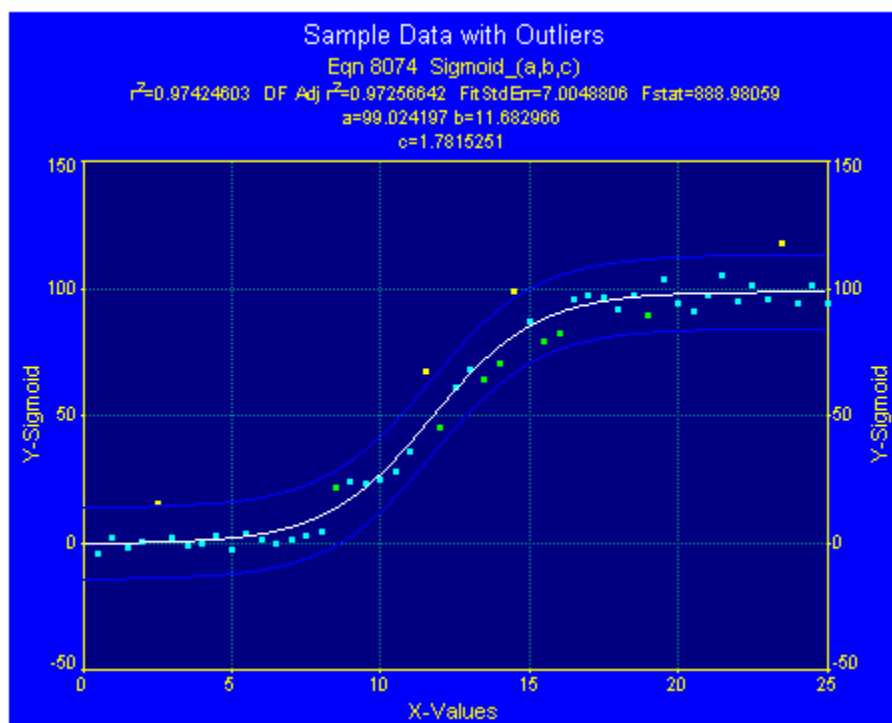
Start With Good Data

The greatest pitfall is probably using invalid data that consist either of erroneous values or some form of noise. An obvious requirement for effective curve-fitting is to have a respectable signal-to-noise ratio. The following data were generated with TableCurve 2D's random number function. The linear trend indicated below by the straight-line fit is completely without merit.

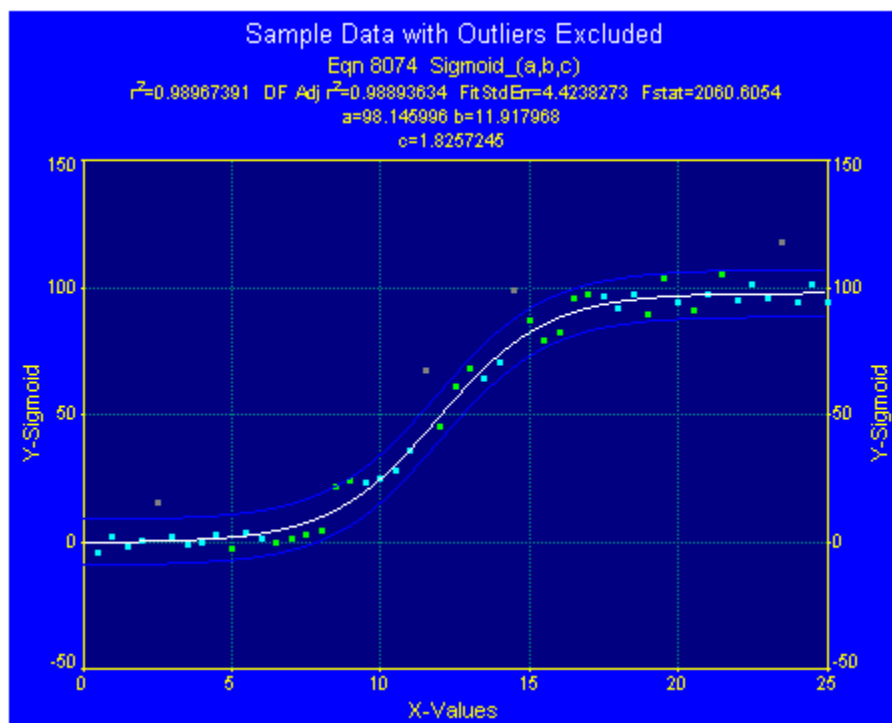


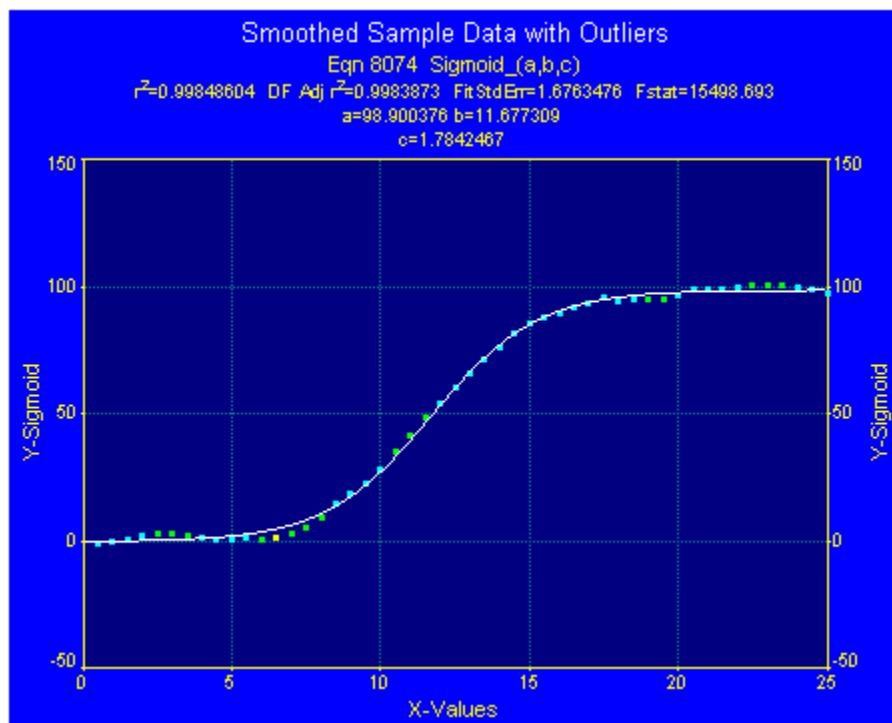
Remove Outliers Rather Than Smoothing

TableCurve 2D's **smoothing and denoising procedures** are often very effective in removing noise from a data set. Smoothing is not, however, a good choice for dealing with outliers. The following data set, when fitted to a sigmoid, has four points outside a 95% prediction interval. The coloring of these points based upon the number of overall standard errors in the residuals likewise indicates outliers.



In the following example, the difference in transition width is greater than 5% between marking these points as inactive to exclude them from the fitting and smoothing the original data in an effort to minimize the impact of these outliers. The correct choice will usually be to exclude outliers from the curve-fitting.



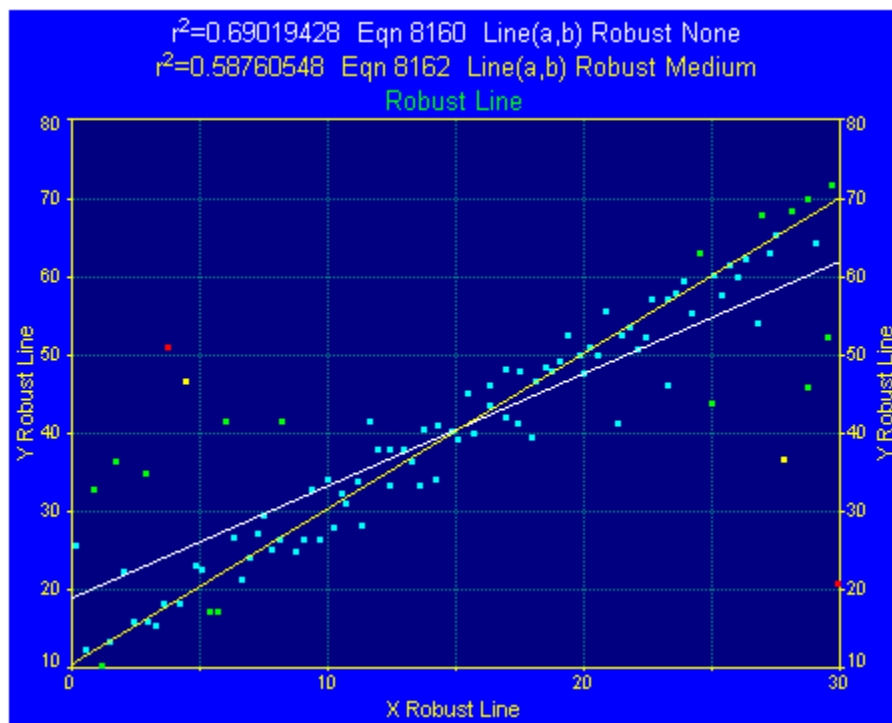


Use Robust Fitting When Appropriate

Use TableCurve 2D's robust fitting procedures when fitting a non-linear model (or are willing to fit a linear model by a UDF), and one or more of the following are true:

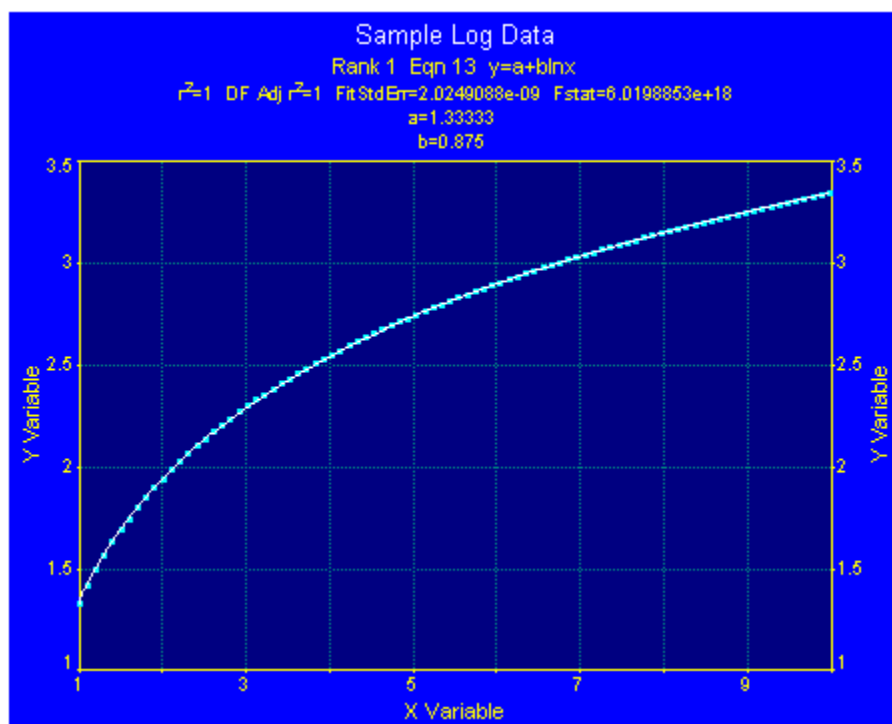
- the data span many orders of magnitude in Y and the low-valued Y points are not factoring into the least-squares fit.
- the distribution of residuals is likely to have broader tails than that of a Gaussian distribution.
- the data are noisy, outliers are suspected, but excluding these outliers is not straightforward.

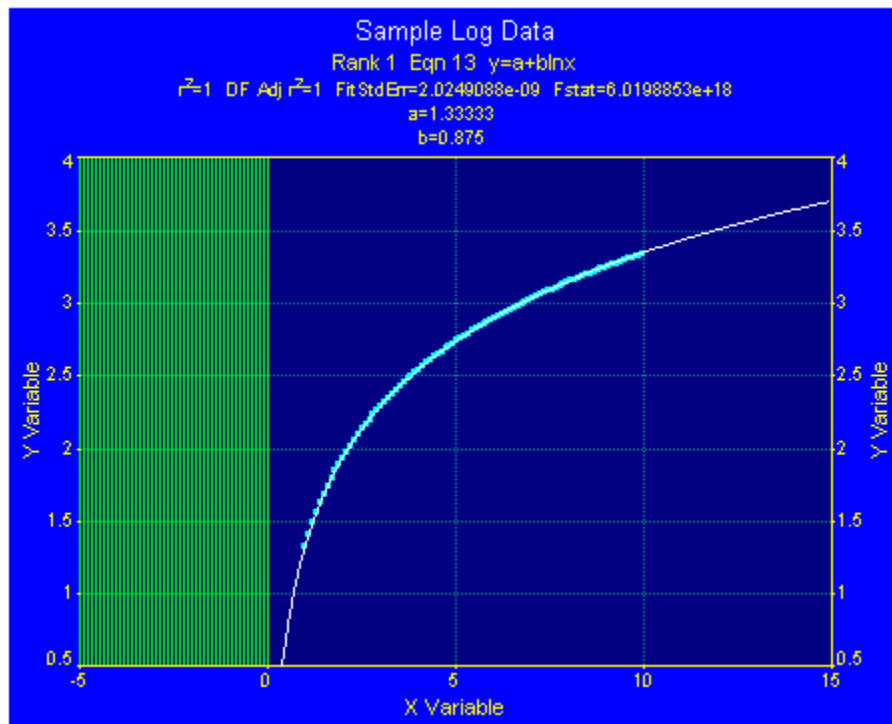
In the following example, noisy straight line data are fitted by least squares and by TableCurve 2D's medium robustness procedure, that which assumes Lorentzian errors. Since robust methods assume a wider distribution of errors, such fits are less impacted by the outliers.



Beware Of Undefined Regions

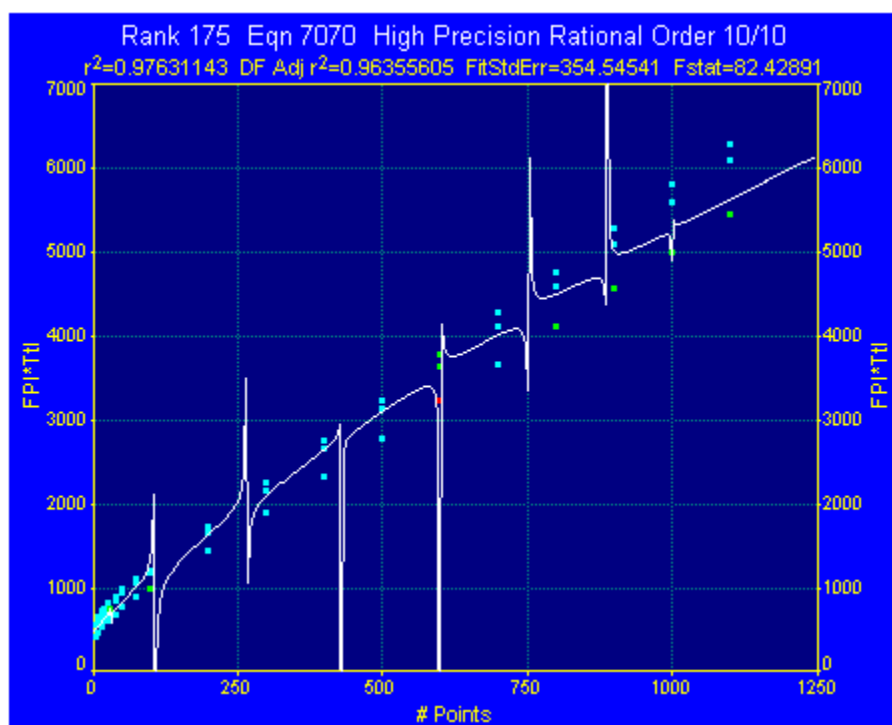
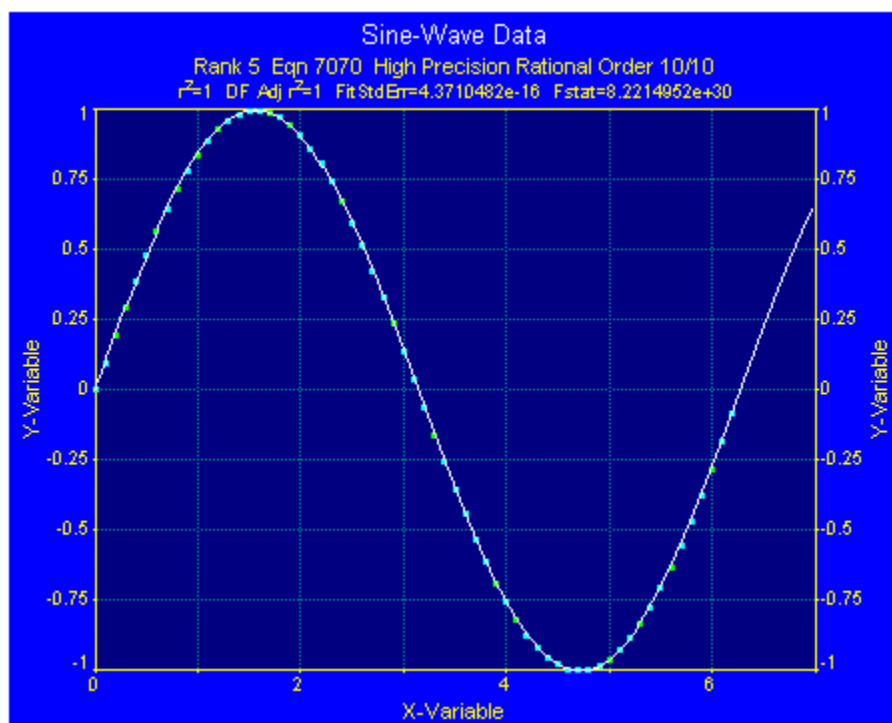
The following data set is fitted perfectly by the basic log equation. Zooming out shows a good chance of extrapolating to higher X-values, but as would be expected, not to values less than 0. Note the vertical bars that indicate math errors for negative X-values. If this equation were selected, the X-values would have to be restricted to values greater than 0.





Beware of Rational Equation Singularities

Many of the special functions offered in numeric packages are rational equations which resulted from fitting tabulated data. This is a testimony to the power of these equations for fitting a diverse spectrum of complex data. All is not paradise, however. Although the top graph shows a 10th/10th order rational function superbly fitting one wavelength of a sine-wave, the bottom graph shows this same 10th/10th order rational failing completely in fitting a simple data set containing multiple Y-observations for each X-value. Note the reasonably high r^2 value in the case with multiple singularities.



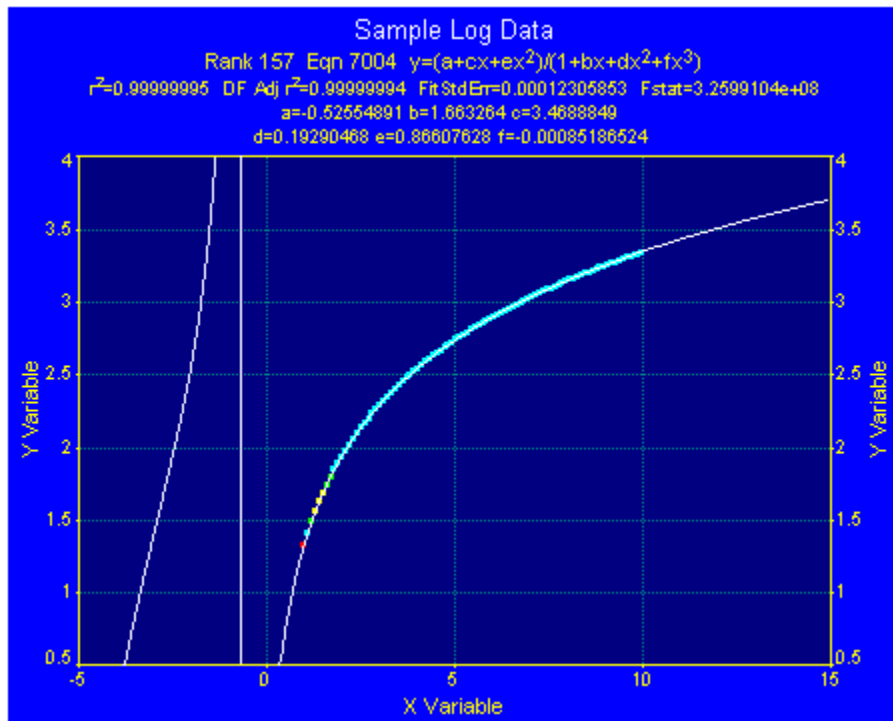
With noisy data, or when there are multiple Y-values for a single X, a rational equation tends to produce its characteristic undefined region(s). Since rationals are quotients of two polynomials, X-values corresponding to the roots of the polynomial in the denominator result in a division by zero thus producing an undefined condition or pole.

TableCurve 2D's equation list incorporates a **Rational Defined Xmin to Xmax** filter. When this filter is checked, the denominator of a rational equations is checked for roots within the X-range of the data. If a

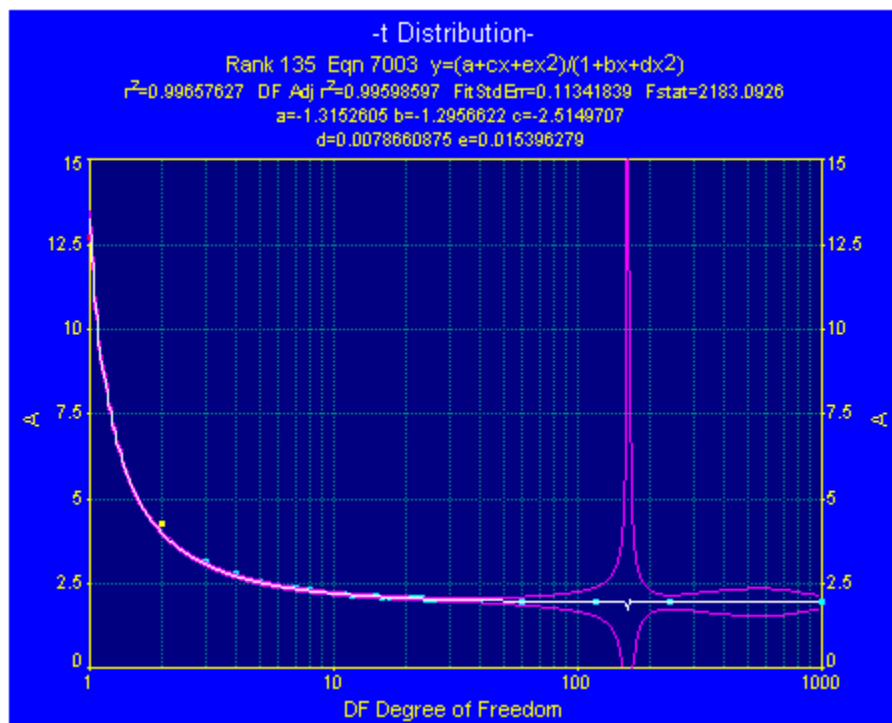
root is found, the equation is filtered from the list.

For rational equations, TableCurve 2D's **Numeric Summary** reports the numeric values for these poles. For denominators that are polynomials in x, all real roots are reported. For other polynomials, a bracketing and root finding are used to detect roots in the range of the data.

Even when there are no singularities in the data range, there is still a strong possibility rational equations will evidence undefined or unstable regions outside the X-data range, as in the graph that follows. If you are using a rational equation for extrapolation, inspect the region of interest very closely, including the confidence intervals.

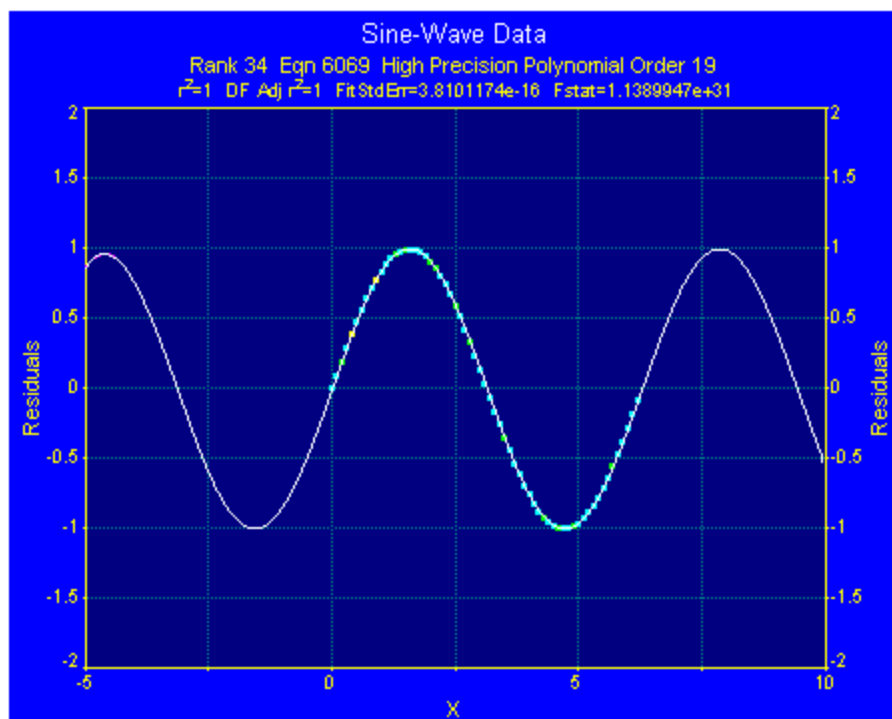


Bear in mind that some poles are so small, they may not be drawn by the curve rendering algorithm. You must thus rely on the numeric report of roots. In those instances where a small blip is observed, such a data region can be shown to be quite unstable by inspecting confidence intervals, as in the graph that follows.

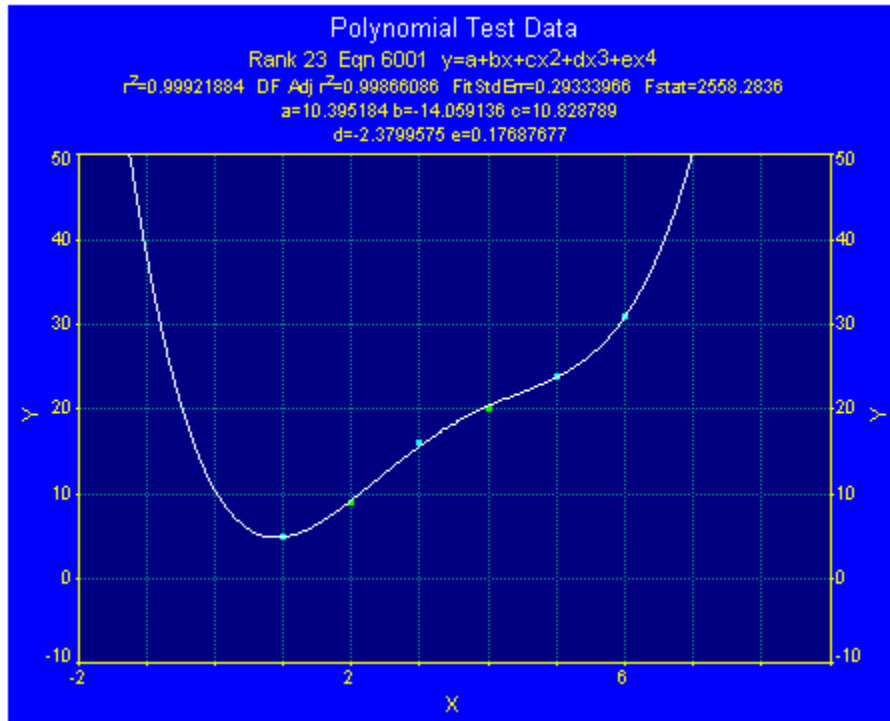


Beware of Extrapolating Polynomial Equations

Polynomials do not suffer the undefined or unstable regions characteristic of rational equations. Polynomials have their own peculiar brand of pitfall, and most often this exists in the area of extrapolations. Relying on polynomials for extrapolations and forecasts can be ruinous. The lure is of a high r^2 fit that manages any number of complex data shapes such as the sine wave below where one wavelength is fitted almost perfectly by a 19th order polynomial.

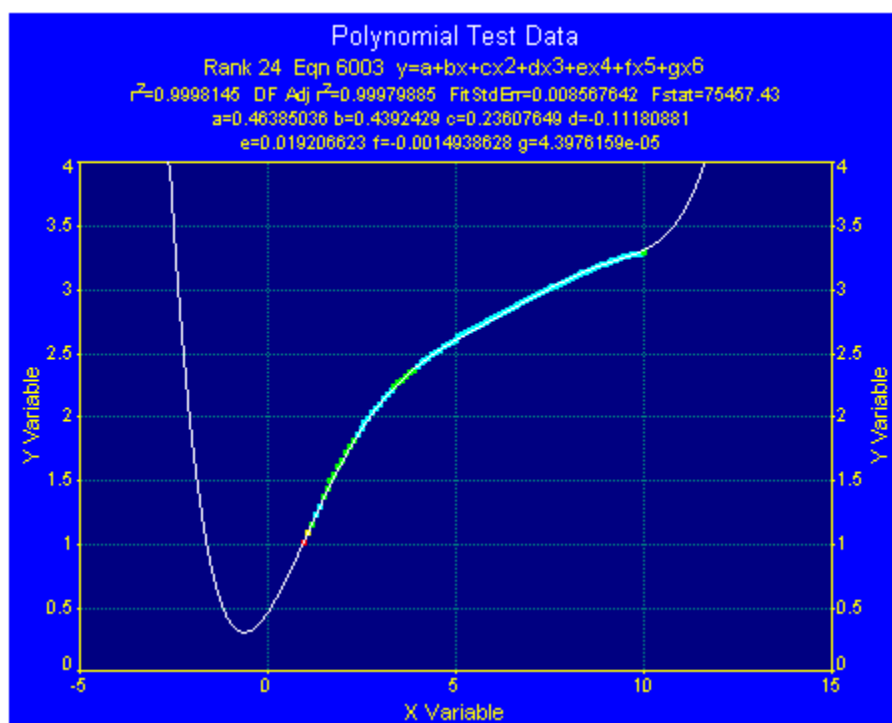
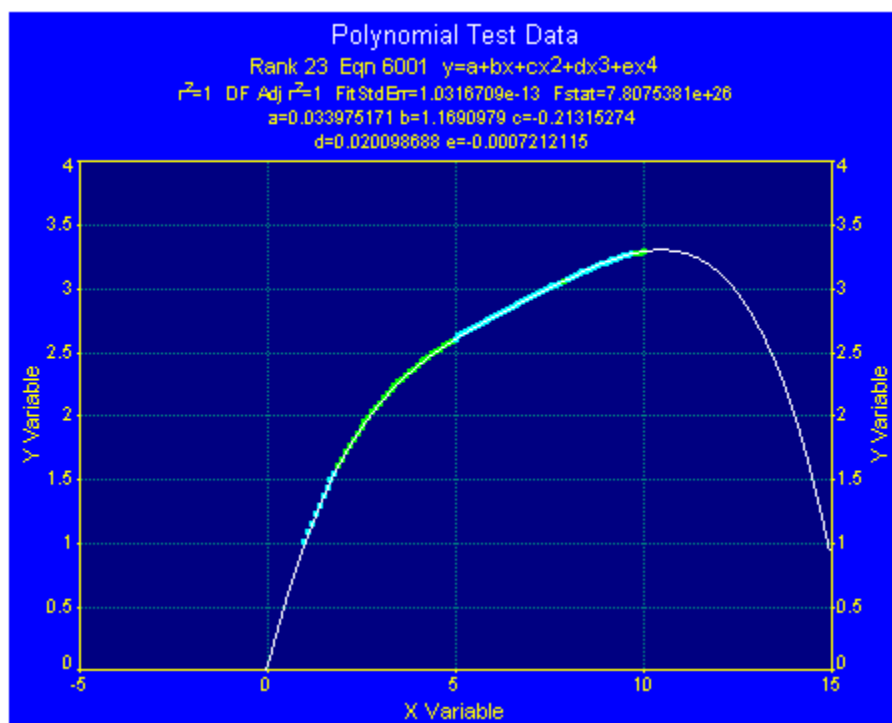


The 4th order polynomial below, however, offers miserably poor extrapolations as well as fitting the noise within the data.



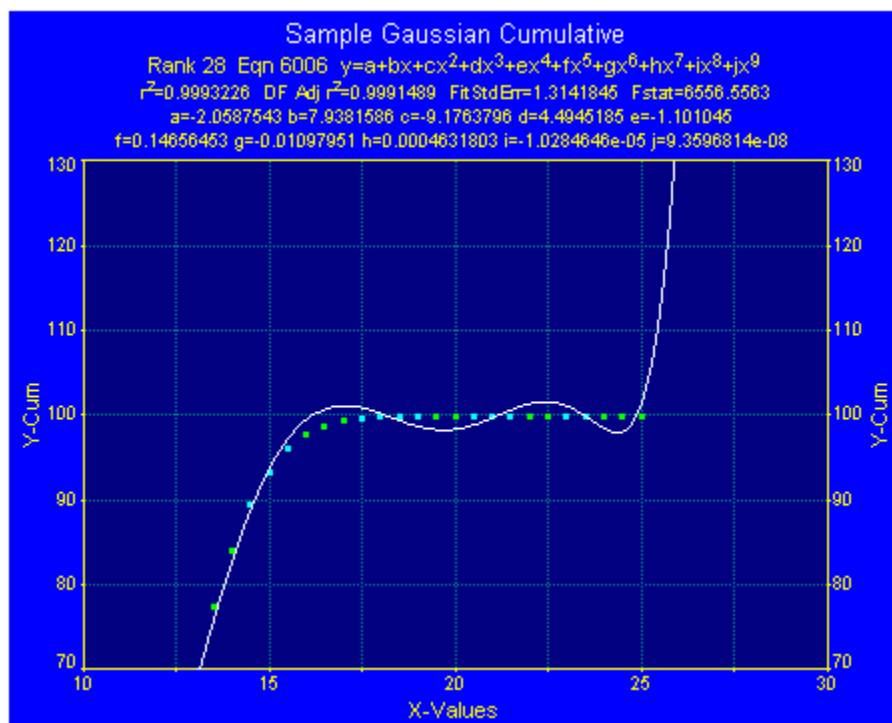
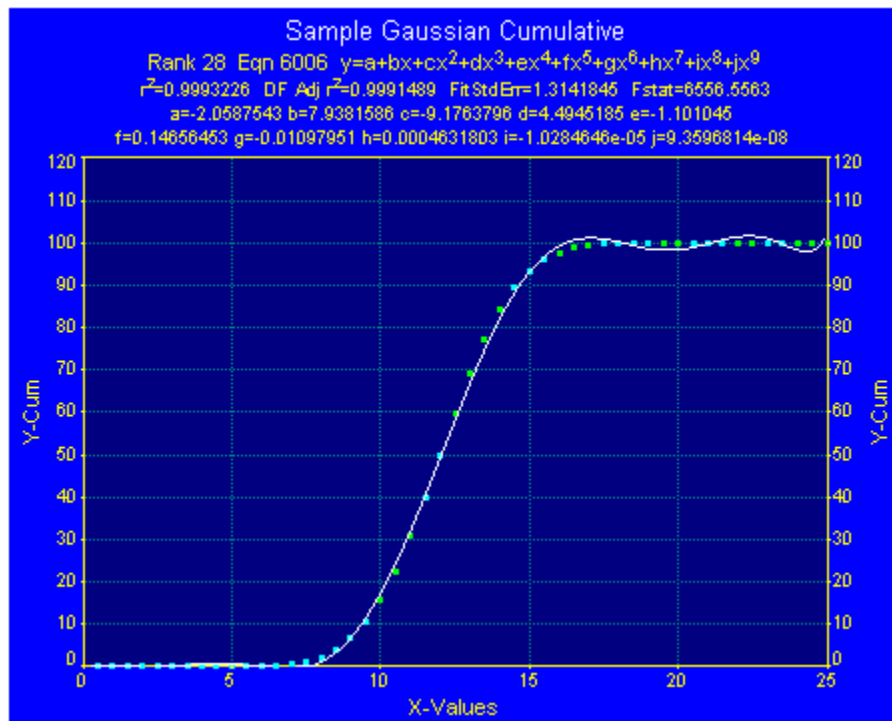
The failed polynomial extrapolations do not even take any consistent directions for a given data set. Both examples that follow are for the same data set.

Note how the 4th order polynomial evidences a very rapid drop in value when extrapolating to higher X-values. The 6th order equation, on the other hand, has exactly the opposite trend for extrapolations to higher X. Extreme caution is thus advised if you intend to use any of the higher order polynomials for extrapolations.

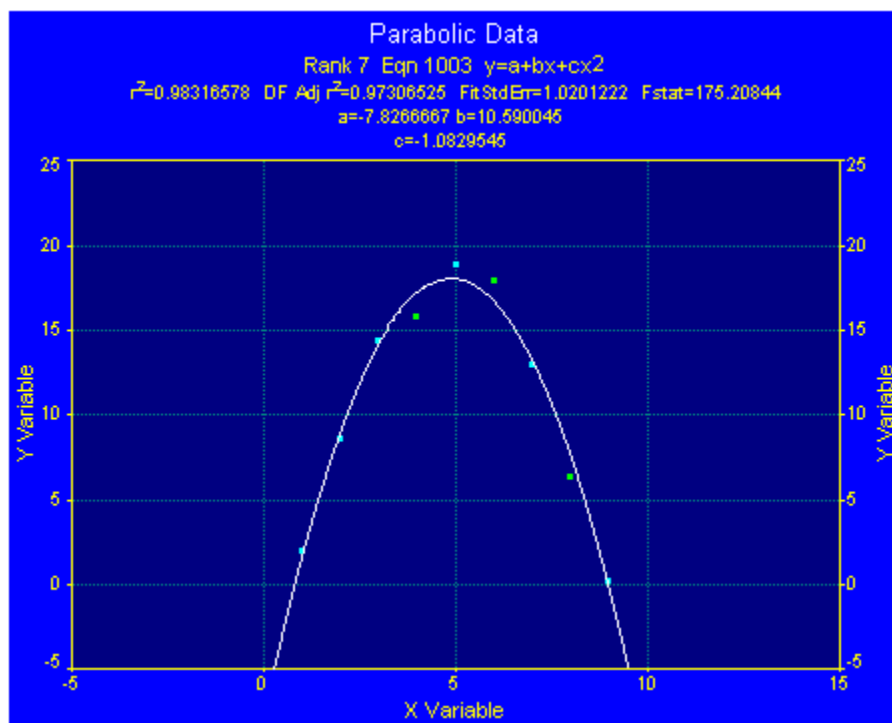
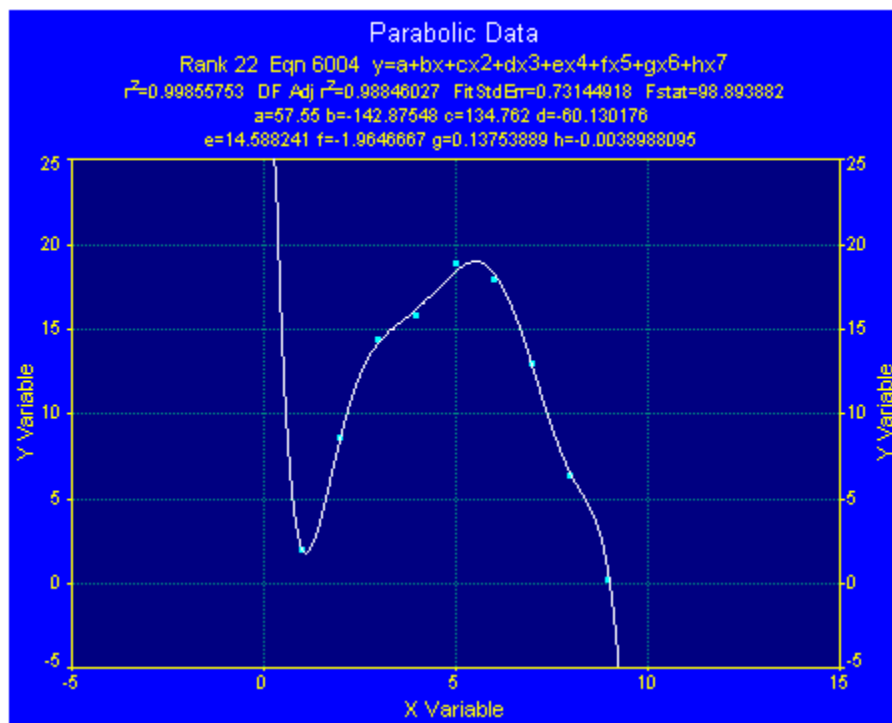


Higher order polynomials tend to do a superb job of fitting curves, but tend to do a very poor job when such curves are combined with relatively straight lines.

In this Gaussian Cumulative data set, the 9th order polynomial has a peculiar instability at the upper plateau. To see such trends more readily, TableCurve 2D offers you to means to zoom-in on graph to see any area of interest.



The most annoying trend of higher order polynomials must certainly be their ability to fit the noise with data. In the following example, noisy parabolic data is fitted to both 7th order and 2nd order polynomials. When fitting polynomials to data, the number of data points should significantly exceed the number of coefficients. Some statistical experts go so far as to recommend that your data point count be at least five to ten times the number of coefficients fitted.

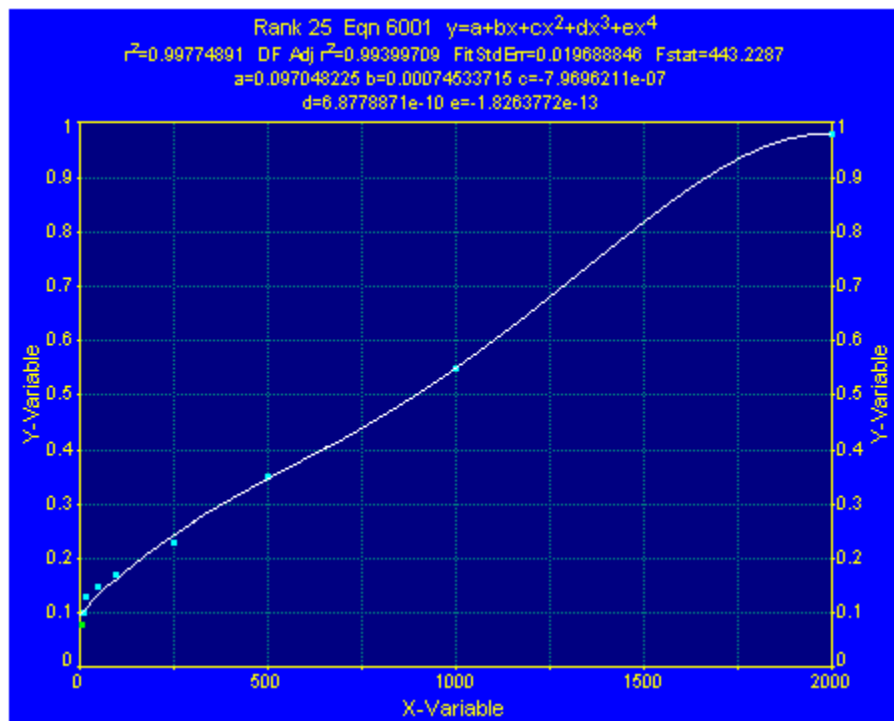


Failure To Examine Data At Low X And Y Values

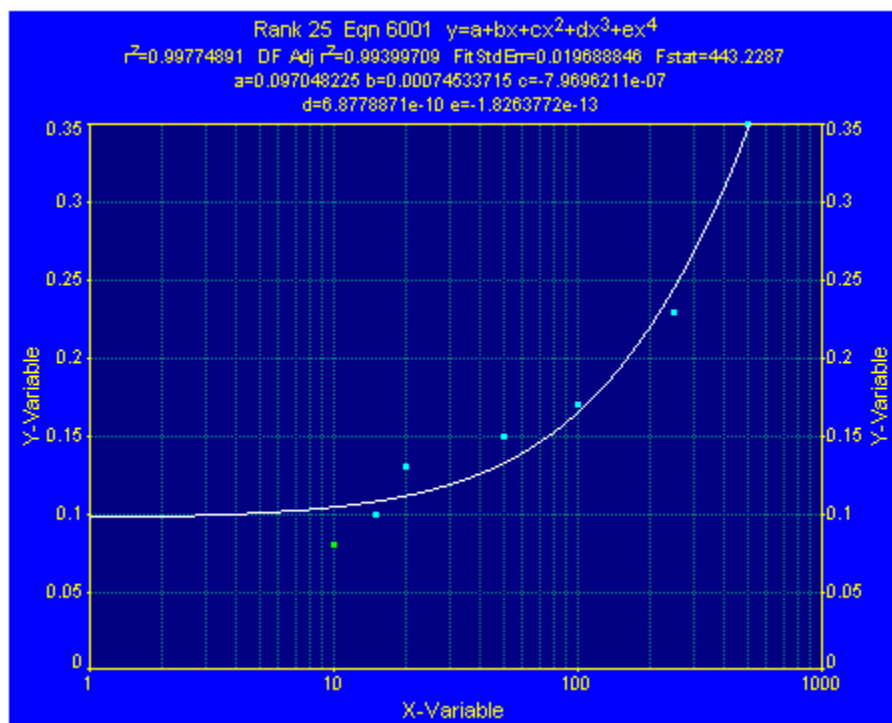
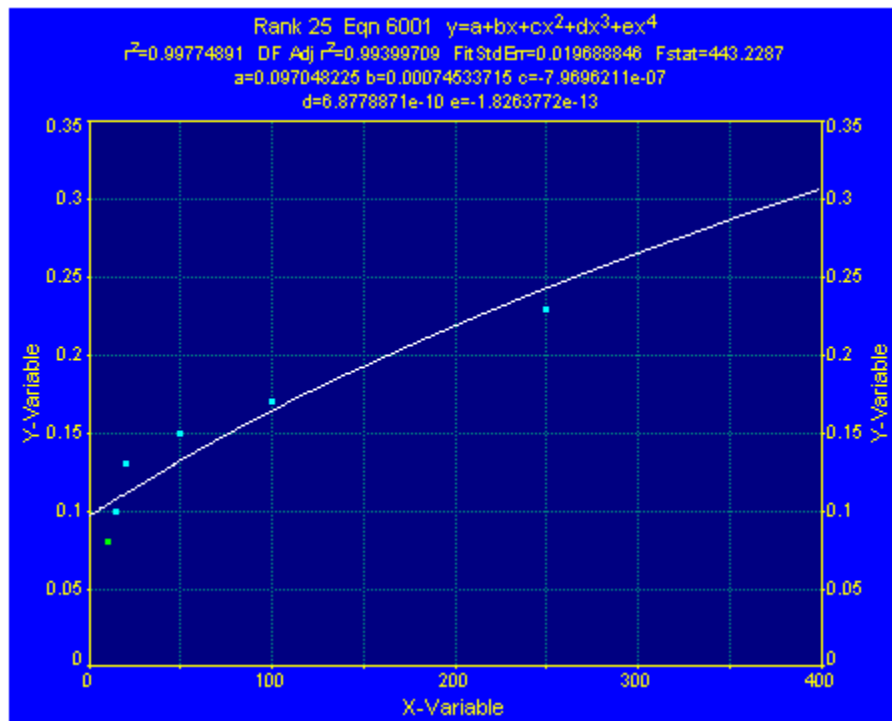
Although there are different ways to calculate a goodness-of-fit criteria, most are strongly sensitive to the magnitude of the differences between the actual and predicted Y-values. This is especially true when the Y-values span more than one order of magnitude. The least-squares method generally uses absolute differences rather than any normalized form, and as such the goodness-of-fit values may not reflect the quality of the curve-fit at this low end of the data range. This is why numerical indices of fit should only be

used only as starting references. The curve-fit in such instances should always be closely examined at low Y values.

When the following data set is fitted to a 4th order polynomial equation, all indications by r^2 and by initial appearance are of a very successful fit. Much of the data, however, is concentrated at low values of Y.



Note how the quality of the curve-fit in this lower region is shown to be less than ideal by first zooming in and then by setting a logarithmic X-axis. The zoomed-in semi-log graph clearly shows a trend in the data that this equation altogether fails to fit. In this instance, a good approach would be to scan the candidate equations with this zoomed-in semi-log scaling until finding one that does fit this trend at the low end of the Y-data range.



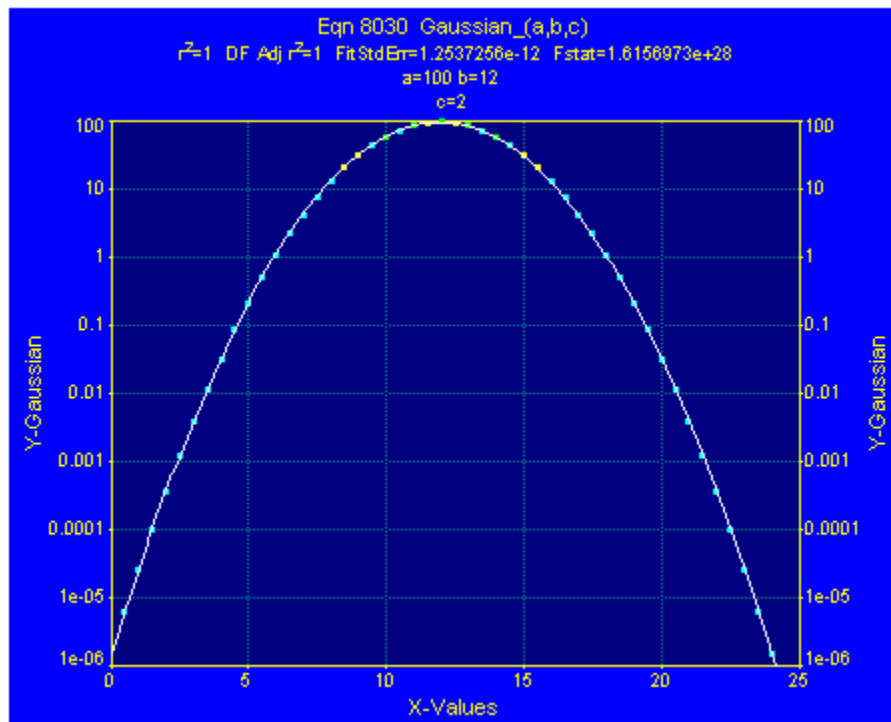
Reversing X And Y Values

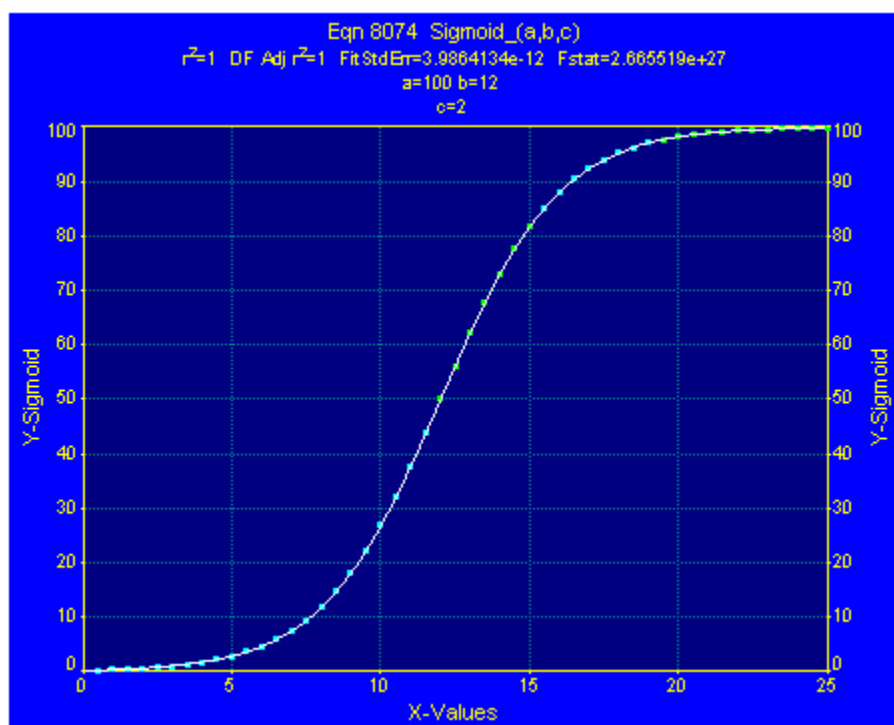
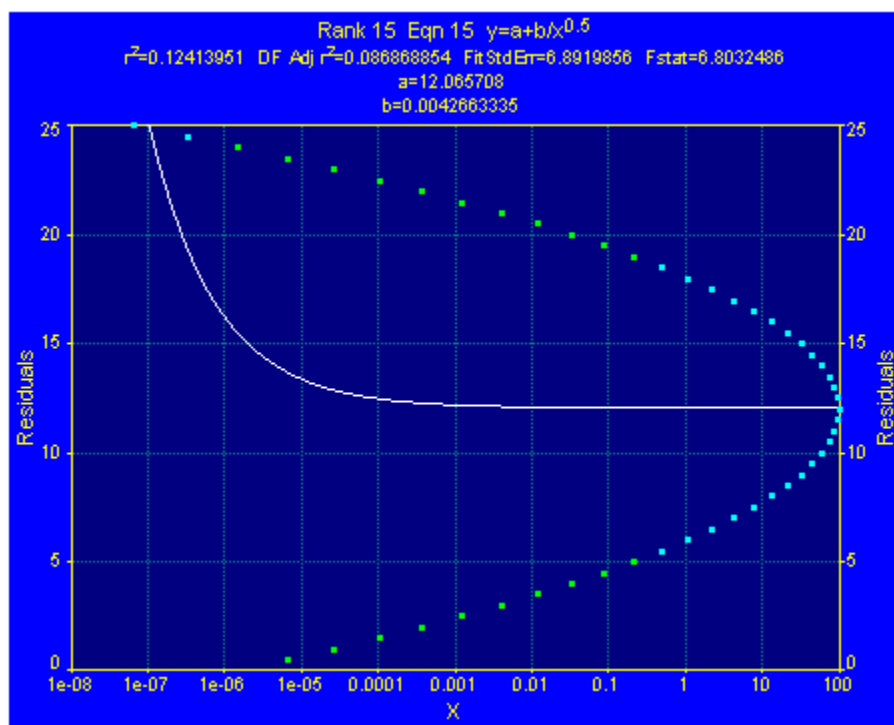
Another pitfall worth noting involves a misunderstanding of the curve-fitting process. Most curve-fitting software is designed to fit data tables in which there is one discrete Y-value for each X-value. If there are more than one Y-value for each X-value, all curves will fit a single Y-value to this X-value. This is seldom a problem with original data, since real-world data usually consists of single observations of X and Y for each data point. When an experiment ramps up in X-values and then back down again, it is common

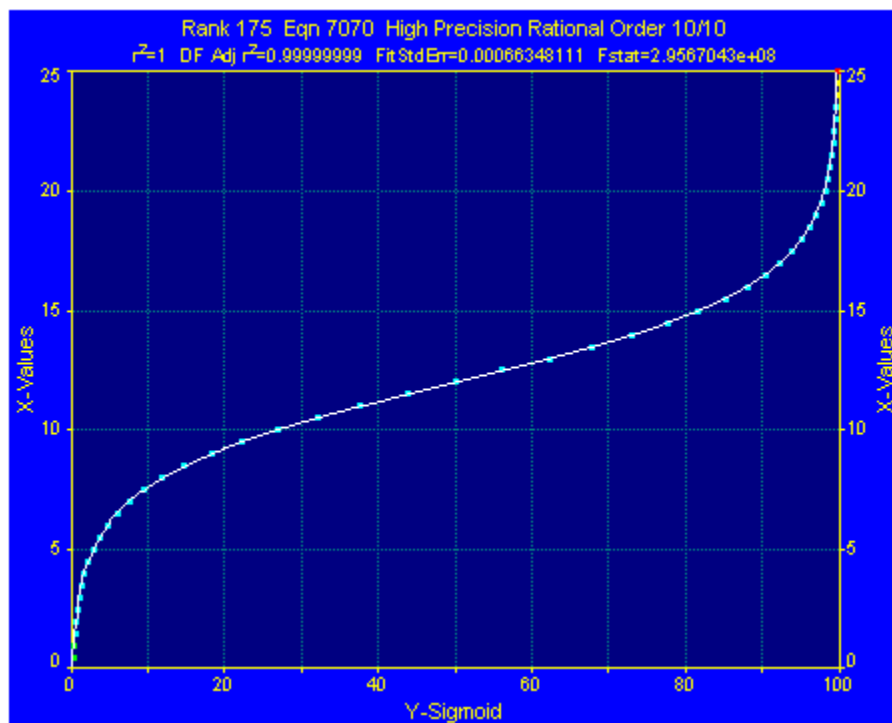
procedure to do separate data analyses for the ascending and descending processes.

The difficulty more commonly appears when X and Y values are reversed in hopes of achieving an equation in which X is given as a function of Y. In programming, such a procedure is far more code-efficient and time-efficient than root finding routines. The pitfall lies in the fact that this reversal will not result in a successful curve-fit equation on certain types of data tables. In particular, y of x must be single valued.

Peak-type data, such as the Gaussian in the first two graphs which follow, is one data type that cannot be reversed. This is also true of waveforms and some parabolic curves. Transition data, such as the sigmoid data in the graphs following the Gaussian, can be successfully reversed. In this case it is successfully fitted by a rational polynomial.







Trusting Numbers Rather Than Your Eyes

Some scientists and engineers developed curve-fitting skills in times when primitive number crunching software was the rule. The results of a curve-fitting effort were always numbers, residual tables, and possibly a simple residuals plot. Many pitfalls existed within this framework simply because so much was hidden. Although numeric goodness-of-fit values are good places to start, the selection of a curve-fit equation should be graphical. TableCurve 2D graphically displays the important aspects of the curve-fit process and encourages you to use your eyes and your judgment to discern the most appropriate model for your data.

Linear Equation List

The TableCurve 2D linear equations are listed by the following categories:

- Eqns 1-105: Single Term Functions with Intercept
- Eqns 1001-1630: Two Term Functions with Intercept
- Eqns 2001-3330: Three Term Functions with Intercept
- Eqns 4001-5140: Four Term Functions with Intercept
- Eqns 6001-6870: Polynomial Equations
- Eqns 7001-7939: Rational Equations

Eqns 1-105: Single Term Functions with Intercept

1 $y = a + bx$	22 $\ln y = a + bx$	43 $\sqrt[y]{y} = a + bx$	64 $\sqrt{y} = a + bx$	85 $y^2 = a + bx$
2 $y = a + bx \ln x$	23 $\ln y = a + bx \ln x$	44 $\sqrt[y]{y} = a + bx \ln x$	65 $\sqrt{y} = a + bx \ln x$	86 $y^2 = a + bx \ln x$
3 $y = a + bx\sqrt{x}$	24 $\ln y = a + bx\sqrt{x}$	45 $\sqrt[y]{y} = a + bx\sqrt{x}$	66 $\sqrt{y} = a + bx\sqrt{x}$	87 $y^2 = a + bx\sqrt{x}$
4 $y = a + bx^2$	25 $\ln y = a + bx^2$	46 $\sqrt[y]{y} = a + bx^2$	67 $\sqrt{y} = a + bx^2$	88 $y^2 = a + bx^2$
5 $y = a + bx^2 \ln x$	26 $\ln y = a + bx^2 \ln x$	47 $\sqrt[y]{y} = a + bx^2 \ln x$	68 $\sqrt{y} = a + bx^2 \ln x$	89 $y^2 = a + bx^2 \ln x$
6 $y = a + bx^2 \sqrt{x}$	27 $\ln y = a + bx^2 \sqrt{x}$	48 $\sqrt[y]{y} = a + bx^2 \sqrt{x}$	69 $\sqrt{y} = a + bx^2 \sqrt{x}$	90 $y^2 = a + bx^2 \sqrt{x}$
7 $y = a + bx^3$	28 $\ln y = a + bx^3$	49 $\sqrt[y]{y} = a + bx^3$	70 $\sqrt{y} = a + bx^3$	91 $y^2 = a + bx^3$
8 $y = a + b e^x$	29 $\ln y = a + b e^x$	50 $\sqrt[y]{y} = a + b e^x$	71 $\sqrt{y} = a + b e^x$	92 $y^2 = a + b e^x$
9 $y = a + b\sqrt{x} \ln x$	30 $\ln y = a + b\sqrt{x} \ln x$	51 $\sqrt[y]{y} = a + b\sqrt{x} \ln x$	72 $\sqrt{y} = a + b\sqrt{x} \ln x$	93 $y^2 = a + b\sqrt{x} \ln x$
10 $y = a + b(\ln x)^2$	31 $\ln y = a + b(\ln x)^2$	52 $\sqrt[y]{y} = a + b(\ln x)^2$	73 $\sqrt{y} = a + b(\ln x)^2$	94 $y^2 = a + b(\ln x)^2$
11 $y = a + b \frac{x}{\ln x}$	32 $\ln y = a + b \frac{x}{\ln x}$	53 $\sqrt[y]{y} = a + b \frac{x}{\ln x}$	74 $\sqrt{y} = a + b \frac{x}{\ln x}$	95 $y^2 = a + b \frac{x}{\ln x}$
12 $y = a + b\sqrt{x}$	33 $\ln y = a + b\sqrt{x}$	54 $\sqrt[y]{y} = a + b\sqrt{x}$	75 $\sqrt{y} = a + b\sqrt{x}$	96 $y^2 = a + b\sqrt{x}$
13 $y = a + b \ln x$	34 $\ln y = a + b \ln x$	55 $\sqrt[y]{y} = a + b \ln x$	76 $\sqrt{y} = a + b \ln x$	97 $y^2 = a + b \ln x$
14 $y = a + \frac{b}{\ln x}$	35 $\ln y = a + \frac{b}{\ln x}$	56 $\sqrt[y]{y} = a + \frac{b}{\ln x}$	77 $\sqrt{y} = a + \frac{b}{\ln x}$	98 $y^2 = a + \frac{b}{\ln x}$
15 $y = a + \frac{b}{\sqrt{x}}$	36 $\ln y = a + \frac{b}{\sqrt{x}}$	57 $\sqrt[y]{y} = a + \frac{b}{\sqrt{x}}$	78 $\sqrt{y} = a + \frac{b}{\sqrt{x}}$	99 $y^2 = a + \frac{b}{\sqrt{x}}$
16 $y = a + b \frac{\ln x}{x}$	37 $\ln y = a + b \frac{\ln x}{x}$	58 $\sqrt[y]{y} = a + b \frac{\ln x}{x}$	79 $\sqrt{y} = a + b \frac{\ln x}{x}$	100 $y^2 = a + b \frac{\ln x}{x}$
17 $y = a + \frac{b}{x}$	38 $\ln y = a + \frac{b}{x}$	59 $\sqrt[y]{y} = a + \frac{b}{x}$	80 $\sqrt{y} = a + \frac{b}{x}$	101 $y^2 = a + \frac{b}{x}$
18 $y = a + \frac{b}{x\sqrt{x}}$	39 $\ln y = a + \frac{b}{x\sqrt{x}}$	60 $\sqrt[y]{y} = a + \frac{b}{x\sqrt{x}}$	81 $\sqrt{y} = a + \frac{b}{x\sqrt{x}}$	102 $y^2 = a + \frac{b}{x\sqrt{x}}$
19 $y = a + b \frac{\ln x}{x^2}$	40 $\ln y = a + b \frac{\ln x}{x^2}$	61 $\sqrt[y]{y} = a + b \frac{\ln x}{x^2}$	82 $\sqrt{y} = a + b \frac{\ln x}{x^2}$	103 $y^2 = a + b \frac{\ln x}{x^2}$
20 $y = a + \frac{b}{x^2}$	41 $\ln y = a + \frac{b}{x^2}$	62 $\sqrt[y]{y} = a + \frac{b}{x^2}$	83 $\sqrt{y} = a + \frac{b}{x^2}$	104 $y^2 = a + \frac{b}{x^2}$
21 $y = a + b e^{-x}$	42 $\ln y = a + b e^{-x}$	63 $\sqrt[y]{y} = a + b e^{-x}$	84 $\sqrt{y} = a + b e^{-x}$	105 $y^2 = a + b e^{-x}$

Eqns 1001-1630: Two Term Functions with Intercept

1001 $y = a + bx + cx \ln x$	1025 $y = a + bx \ln x + cx^3$	1049 $y = a + bx\sqrt{x} + c \ln x$	1073 $y = a + bx^2 + \frac{c}{x^2}$
1002 $y = a + bx + cx\sqrt{x}$	1026 $y = a + bx \ln x + c e^x$	1050 $y = a + bx\sqrt{x} + \frac{c}{\ln x}$	1074 $y = a + bx^2 + c e^{-x}$
1003 $y = a + bx + cx^2$	1027 $y = a + bx \ln x + c\sqrt{x} \ln x$	1051 $y = a + bx\sqrt{x} + \frac{c}{\sqrt{x}}$	1075 $y = a + bx^2 \ln x + cx^2 \sqrt{x}$
1004 $y = a + bx + cx^2 \ln x$	1028 $y = a + bx \ln x + c(\ln x)^2$	1052 $y = a + bx\sqrt{x} + c \frac{\ln x}{x}$	1076 $y = a + bx^2 \ln x + cx^3$
1005 $y = a + bx + cx^2 \sqrt{x}$	1029 $y = a + bx \ln x + c \frac{x}{\ln x}$	1053 $y = a + bx\sqrt{x} + \frac{c}{x}$	1077 $y = a + bx^2 \ln x + c e^x$
1006 $y = a + bx + cx^3$	1030 $y = a + bx \ln x + c\sqrt{x}$	1054 $y = a + bx\sqrt{x} + \frac{c}{x\sqrt{x}}$	1078 $y = a + bx^2 \ln x + c\sqrt{x} \ln x$
1007 $y = a + bx + c e^x$	1031 $y = a + bx \ln x + c \ln x$	1055 $y = a + bx\sqrt{x} + c \frac{\ln x}{x^2}$	1079 $y = a + bx^2 \ln x + c(\ln x)^2$
1008 $y = a + bx + c\sqrt{x} \ln x$	1032 $y = a + bx \ln x + \frac{c}{\ln x}$	1056 $y = a + bx\sqrt{x} + \frac{c}{x^2}$	1080 $y = a + bx^2 \ln x + c \frac{x}{\ln x}$
1009 $y = a + bx + c(\ln x)^2$	1033 $y = a + bx \ln x + \frac{c}{\sqrt{x}}$	1057 $y = a + bx\sqrt{x} + c e^{-x}$	1081 $y = a + bx^2 \ln x + c\sqrt{x}$
1010 $y = a + bx + c \frac{x}{\ln x}$	1034 $y = a + bx \ln x + c \frac{\ln x}{x}$	1058 $y = a + bx^2 + cx^2 \ln x$	1082 $y = a + bx^2 \ln x + c \ln x$
1011 $y = a + bx + c\sqrt{x}$	1035 $y = a + bx \ln x + \frac{c}{x}$	1059 $y = a + bx^2 + cx^2 \sqrt{x}$	1083 $y = a + bx^2 \ln x + \frac{c}{\ln x}$
1012 $y = a + bx + c \ln x$	1036 $y = a + bx \ln x + \frac{c}{x\sqrt{x}}$	1060 $y = a + bx^2 + cx^3$	1084 $y = a + bx^2 \ln x + \frac{c}{\sqrt{x}}$
1013 $y = a + bx + \frac{c}{\ln x}$	1037 $y = a + bx \ln x + c \frac{\ln x}{x^2}$	1061 $y = a + bx^2 + c e^x$	1085 $y = a + bx^2 \ln x + c \frac{\ln x}{x}$
1014 $y = a + bx + \frac{c}{\sqrt{x}}$	1038 $y = a + bx \ln x + \frac{c}{x^2}$	1062 $y = a + bx^2 + c\sqrt{x} \ln x$	1086 $y = a + bx^2 \ln x + \frac{c}{x}$
1015 $y = a + bx + c \frac{\ln x}{x}$	1039 $y = a + bx \ln x + c e^{-x}$	1063 $y = a + bx^2 + c(\ln x)^2$	1087 $y = a + bx^2 \ln x + \frac{c}{x\sqrt{x}}$
1016 $y = a + bx + \frac{c}{x}$	1040 $y = a + bx\sqrt{x} + cx^2$	1064 $y = a + bx^2 + c \frac{x}{\ln x}$	1088 $y = a + bx^2 \ln x + c \frac{\ln x}{x^2}$
1017 $y = a + bx + \frac{c}{x\sqrt{x}}$	1041 $y = a + bx\sqrt{x} + cx^2 \ln x$	1065 $y = a + bx^2 + c\sqrt{x}$	1089 $y = a + bx^2 \ln x + \frac{c}{x^2}$
1018 $y = a + bx + c \frac{\ln x}{x^2}$	1042 $y = a + bx\sqrt{x} + cx^2 \sqrt{x}$	1066 $y = a + bx^2 + c \ln x$	1090 $y = a + bx^2 \ln x + c e^{-x}$
1019 $y = a + bx + \frac{c}{x^2}$	1043 $y = a + bx\sqrt{x} + cx^3$	1067 $y = a + bx^2 + \frac{c}{\ln x}$	1091 $y = a + bx^2 \sqrt{x} + cx^3$
1020 $y = a + bx + c e^{-x}$	1044 $y = a + bx\sqrt{x} + c e^x$	1068 $y = a + bx^2 + \frac{c}{\sqrt{x}}$	1092 $y = a + bx^2 \sqrt{x} + c e^x$
1021 $y = a + bx \ln x + cx\sqrt{x}$	1045 $y = a + bx\sqrt{x} + c\sqrt{x} \ln x$	1069 $y = a + bx^2 + c \frac{\ln x}{x}$	1093 $y = a + bx^2 \sqrt{x} + c\sqrt{x} \ln x$
1022 $y = a + bx \ln x + cx^2$	1046 $y = a + bx\sqrt{x} + c(\ln x)^2$	1070 $y = a + bx^2 + \frac{c}{x}$	1094 $y = a + bx^2 \sqrt{x} + c(\ln x)^2$
1023 $y = a + bx \ln x + cx^2 \ln x$	1047 $y = a + bx\sqrt{x} + c \frac{x}{\ln x}$	1071 $y = a + bx^2 + \frac{c}{x\sqrt{x}}$	1095 $y = a + bx^2 \sqrt{x} + c \frac{x}{\ln x}$
1024 $y = a + bx \ln x + cx^2 \sqrt{x}$	1048 $y = a + bx\sqrt{x} + c\sqrt{x}$	1072 $y = a + bx^2 + c \frac{\ln x}{x^2}$	1096 $y = a + bx^2 \sqrt{x} + c\sqrt{x}$

1097 $y = a + bx^2\sqrt{x} + c \ln x$	1122 $y = a + b e^x + c \frac{x}{\ln x}$	1147 $y = a + b(\ln x)^2 + c \ln x$	1172 $y = a + b\sqrt{x} + c \frac{\ln x}{x^2}$
1098 $y = a + bx^2\sqrt{x} + \frac{c}{\ln x}$	1123 $y = a + b e^x + c\sqrt{x}$	1148 $y = a + b(\ln x)^2 + \frac{c}{\ln x}$	1173 $y = a + b\sqrt{x} + \frac{c}{x^2}$
1099 $y = a + bx^2\sqrt{x} + \frac{c}{\sqrt{x}}$	1124 $y = a + b e^x + c \ln x$	1149 $y = a + b(\ln x)^2 + \frac{c}{\sqrt{x}}$	1174 $y = a + b\sqrt{x} + c e^{-x}$
1100 $y = a + bx^2\sqrt{x} + c \frac{\ln x}{x}$	1125 $y = a + b e^x + \frac{c}{\ln x}$	1150 $y = a + b(\ln x)^2 + c \frac{\ln x}{x}$	1175 $y = a + b \ln x + \frac{c}{\ln x}$
1101 $y = a + bx^2\sqrt{x} + \frac{c}{x}$	1126 $y = a + b e^x + \frac{c}{\sqrt{x}}$	1151 $y = a + b(\ln x)^2 + \frac{c}{x}$	1176 $y = a + b \ln x + \frac{c}{\sqrt{x}}$
1102 $y = a + bx^2\sqrt{x} + \frac{c}{x\sqrt{x}}$	1127 $y = a + b e^x + c \frac{\ln x}{x}$	1152 $y = a + b(\ln x)^2 + \frac{c}{x\sqrt{x}}$	1177 $y = a + b \ln x + c \frac{\ln x}{x}$
1103 $y = a + bx^2\sqrt{x} + c \frac{\ln x}{x^2}$	1128 $y = a + b e^x + \frac{c}{x}$	1153 $y = a + b(\ln x)^2 + c \frac{\ln x}{x^2}$	1178 $y = a + b \ln x + \frac{c}{x}$
1104 $y = a + bx^2\sqrt{x} + \frac{c}{x^2}$	1129 $y = a + b e^x + \frac{c}{x\sqrt{x}}$	1154 $y = a + b(\ln x)^2 + \frac{c}{x^2}$	1179 $y = a + b \ln x + \frac{c}{x\sqrt{x}}$
1105 $y = a + bx^2\sqrt{x} + c e^{-x}$	1130 $y = a + b e^x + c \frac{\ln x}{x^2}$	1155 $y = a + b(\ln x)^2 + c e^{-x}$	1180 $y = a + b \ln x + c \frac{\ln x}{x^2}$
1106 $y = a + bx^3 + c e^x$	1131 $y = a + b e^x + \frac{c}{x^2}$	1156 $y = a + b \frac{x}{\ln x} + c\sqrt{x}$	1181 $y = a + b \ln x + \frac{c}{x^2}$
1107 $y = a + bx^3 + c\sqrt{x} \ln x$	1132 $y = a + b e^x + c e^{-x}$	1157 $y = a + b \frac{x}{\ln x} + c \ln x$	1182 $y = a + b \ln x + c e^{-x}$
1108 $y = a + bx^3 + c(\ln x)^2$	1133 $y = a + b\sqrt{x} \ln x + c(\ln x)^2$	1158 $y = a + b \frac{x}{\ln x} + \frac{c}{\ln x}$	1183 $y = a + \frac{b}{\ln x} + \frac{c}{\sqrt{x}}$
1109 $y = a + bx^3 + c \frac{x}{\ln x}$	1134 $y = a + b\sqrt{x} \ln x + c \frac{x}{\ln x}$	1159 $y = a + b \frac{x}{\ln x} + \frac{c}{\sqrt{x}}$	1184 $y = a + \frac{b}{\ln x} + c \frac{\ln x}{x}$
1110 $y = a + bx^3 + c\sqrt{x}$	1135 $y = a + b\sqrt{x} \ln x + c\sqrt{x}$	1160 $y = a + b \frac{x}{\ln x} + c \frac{\ln x}{x}$	1185 $y = a + \frac{b}{\ln x} + \frac{c}{x}$
1111 $y = a + bx^3 + c \ln x$	1136 $y = a + b\sqrt{x} \ln x + c \ln x$	1161 $y = a + b \frac{x}{\ln x} + \frac{c}{x}$	1186 $y = a + \frac{b}{\ln x} + \frac{c}{x\sqrt{x}}$
1112 $y = a + bx^3 + \frac{c}{\ln x}$	1137 $y = a + b\sqrt{x} \ln x + \frac{c}{\ln x}$	1162 $y = a + b \frac{x}{\ln x} + \frac{c}{x\sqrt{x}}$	1187 $y = a + \frac{b}{\ln x} + c \frac{\ln x}{x^2}$
1113 $y = a + bx^3 + \frac{c}{\sqrt{x}}$	1138 $y = a + b\sqrt{x} \ln x + \frac{c}{\sqrt{x}}$	1163 $y = a + b \frac{x}{\ln x} + c \frac{\ln x}{x^2}$	1188 $y = a + \frac{b}{\ln x} + \frac{c}{x^2}$
1114 $y = a + bx^3 + c \frac{\ln x}{x}$	1139 $y = a + b\sqrt{x} \ln x + c \frac{\ln x}{x}$	1164 $y = a + b \frac{x}{\ln x} + \frac{c}{x^2}$	1189 $y = a + \frac{b}{\ln x} + c e^{-x}$
1115 $y = a + bx^3 + \frac{c}{x}$	1140 $y = a + b\sqrt{x} \ln x + \frac{c}{x}$	1165 $y = a + b \frac{x}{\ln x} + c e^{-x}$	1190 $y = a + \frac{b}{\sqrt{x}} + c \frac{\ln x}{x}$
1116 $y = a + bx^3 + \frac{c}{x\sqrt{x}}$	1141 $y = a + b\sqrt{x} \ln x + \frac{c}{x\sqrt{x}}$	1166 $y = a + b\sqrt{x} + c \ln x$	1191 $y = a + \frac{b}{\sqrt{x}} + \frac{c}{x}$
1117 $y = a + bx^3 + c \frac{\ln x}{x^2}$	1142 $y = a + b\sqrt{x} \ln x + c \frac{\ln x}{x^2}$	1167 $y = a + b\sqrt{x} + \frac{c}{\ln x}$	1192 $y = a + \frac{b}{\sqrt{x}} + \frac{c}{x\sqrt{x}}$
1118 $y = a + bx^3 + \frac{c}{x^2}$	1143 $y = a + b\sqrt{x} \ln x + \frac{c}{x^2}$	1168 $y = a + b\sqrt{x} + \frac{c}{\sqrt{x}}$	1193 $y = a + \frac{b}{\sqrt{x}} + c \frac{\ln x}{x^2}$
1119 $y = a + bx^3 + c e^{-x}$	1144 $y = a + b\sqrt{x} \ln x + c e^{-x}$	1169 $y = a + b\sqrt{x} + c \frac{\ln x}{x}$	1194 $y = a + \frac{b}{\sqrt{x}} + \frac{c}{x^2}$
1120 $y = a + b e^x + c\sqrt{x} \ln x$	1145 $y = a + b(\ln x)^2 + c \frac{x}{\ln x}$	1170 $y = a + b\sqrt{x} + \frac{c}{x}$	1195 $y = a + \frac{b}{\sqrt{x}} + c e^{-x}$
1121 $y = a + b e^x + c(\ln x)^2$	1146 $y = a + b(\ln x)^2 + c\sqrt{x}$	1171 $y = a + b\sqrt{x} + \frac{c}{x\sqrt{x}}$	1196 $y = a + b \frac{\ln x}{x} + \frac{c}{x}$

1197 $y = a + b \frac{\ln x}{x} + \frac{c}{x\sqrt{x}}$	1222 $\ln y = a + bx + c \ln x$	1247 $\ln y = a + bx \ln x + c \frac{\ln x}{x^2}$	1272 $\ln y = a + bx^2 + c\sqrt{x} \ln x$
1198 $y = a + b \frac{\ln x}{x} + c \frac{\ln x}{x^2}$	1223 $\ln y = a + bx + \frac{c}{\ln x}$	1248 $\ln y = a + bx \ln x + \frac{c}{x^2}$	1273 $\ln y = a + bx^2 + c(\ln x)^2$
1199 $y = a + b \frac{\ln x}{x} + \frac{c}{x^2}$	1224 $\ln y = a + bx + \frac{c}{\sqrt{x}}$	1249 $\ln y = a + bx \ln x + c e^{-x}$	1274 $\ln y = a + bx^2 + c \frac{x}{\ln x}$
1200 $y = a + b \frac{\ln x}{x} + c e^{-x}$	1225 $\ln y = a + bx + c \frac{\ln x}{x}$	1250 $\ln y = a + bx\sqrt{x} + cx^2$	1275 $\ln y = a + bx^2 + c\sqrt{x}$
1201 $y = a + \frac{b}{x} + \frac{c}{x\sqrt{x}}$	1226 $\ln y = a + bx + \frac{c}{x}$	1251 $\ln y = a + bx\sqrt{x} + cx^2 \ln x$	1276 $\ln y = a + bx^2 + c \ln x$
1202 $y = a + \frac{b}{x} + c \frac{\ln x}{x^2}$	1227 $\ln y = a + bx + \frac{c}{x\sqrt{x}}$	1252 $\ln y = a + bx\sqrt{x} + cx^2 \sqrt{x}$	1277 $\ln y = a + bx^2 + \frac{c}{\ln x}$
1203 $y = a + \frac{b}{x} + \frac{c}{x^2}$	1228 $\ln y = a + bx + c \frac{\ln x}{x^2}$	1253 $\ln y = a + bx\sqrt{x} + cx^3$	1278 $\ln y = a + bx^2 + \frac{c}{\sqrt{x}}$
1204 $y = a + \frac{b}{x} + c e^{-x}$	1229 $\ln y = a + bx + \frac{c}{x^2}$	1254 $\ln y = a + bx\sqrt{x} + c e^x$	1279 $\ln y = a + bx^2 + c \frac{\ln x}{x}$
1205 $y = a + \frac{b}{x\sqrt{x}} + c \frac{\ln x}{x^2}$	1230 $\ln y = a + bx + c e^{-x}$	1255 $\ln y = a + bx\sqrt{x} + c\sqrt{x} \ln x$	1280 $\ln y = a + bx^2 + \frac{c}{x}$
1206 $y = a + \frac{b}{x\sqrt{x}} + \frac{c}{x^2}$	1231 $\ln y = a + bx \ln x + cx\sqrt{x}$	1256 $\ln y = a + bx\sqrt{x} + c(\ln x)^2$	1281 $\ln y = a + bx^2 + \frac{c}{x\sqrt{x}}$
1207 $y = a + \frac{b}{x\sqrt{x}} + c e^{-x}$	1232 $\ln y = a + bx \ln x + cx^2$	1257 $\ln y = a + bx\sqrt{x} + c \frac{x}{\ln x}$	1282 $\ln y = a + bx^2 + c \frac{\ln x}{x^2}$
1208 $y = a + b \frac{\ln x}{x^2} + \frac{c}{x^2}$	1233 $\ln y = a + bx \ln x + cx^2 \ln x$	1258 $\ln y = a + bx\sqrt{x} + c\sqrt{x}$	1283 $\ln y = a + bx^2 + \frac{c}{x^2}$
1209 $y = a + b \frac{\ln x}{x^2} + c e^{-x}$	1234 $\ln y = a + bx \ln x + cx^2 \sqrt{x}$	1259 $\ln y = a + bx\sqrt{x} + c \ln x$	1284 $\ln y = a + bx^2 + c e^{-x}$
1210 $y = a + \frac{b}{x^2} + c e^{-x}$	1235 $\ln y = a + bx \ln x + cx^3$	1260 $\ln y = a + bx\sqrt{x} + \frac{c}{\ln x}$	1285 $\ln y = a + bx^2 \ln x + cx^2 \sqrt{x}$
1211 $\ln y = a + bx + cx \ln x$	1236 $\ln y = a + bx \ln x + c e^x$	1261 $\ln y = a + bx\sqrt{x} + \frac{c}{\sqrt{x}}$	1286 $\ln y = a + bx^2 \ln x + cx^3$
1212 $\ln y = a + bx + cx\sqrt{x}$	1237 $\ln y = a + bx \ln x + c\sqrt{x} \ln x$	1262 $\ln y = a + bx\sqrt{x} + c \frac{\ln x}{x}$	1287 $\ln y = a + bx^2 \ln x + c e^x$
1213 $\ln y = a + bx + cx^2$	1238 $\ln y = a + bx \ln x + c(\ln x)^2$	1263 $\ln y = a + bx\sqrt{x} + \frac{c}{x}$	1288 $\ln y = a + bx^2 \ln x + c\sqrt{x} \ln x$
1214 $\ln y = a + bx + cx^2 \ln x$	1239 $\ln y = a + bx \ln x + c \frac{x}{\ln x}$	1264 $\ln y = a + bx\sqrt{x} + \frac{c}{x\sqrt{x}}$	1289 $\ln y = a + bx^2 \ln x + c(\ln x)^2$
1215 $\ln y = a + bx + cx^2 \sqrt{x}$	1240 $\ln y = a + bx \ln x + c\sqrt{x}$	1265 $\ln y = a + bx\sqrt{x} + c \frac{\ln x}{x^2}$	1290 $\ln y = a + bx^2 \ln x + c \frac{x}{\ln x}$
1216 $\ln y = a + bx + cx^3$	1241 $\ln y = a + bx \ln x + c \ln x$	1266 $\ln y = a + bx\sqrt{x} + \frac{c}{x^2}$	1291 $\ln y = a + bx^2 \ln x + c\sqrt{x}$
1217 $\ln y = a + bx + c e^x$	1242 $\ln y = a + bx \ln x + \frac{c}{\ln x}$	1267 $\ln y = a + bx\sqrt{x} + c e^{-x}$	1292 $\ln y = a + bx^2 \ln x + c \ln x$
1218 $\ln y = a + bx + c\sqrt{x} \ln x$	1243 $\ln y = a + bx \ln x + \frac{c}{\sqrt{x}}$	1268 $\ln y = a + bx^2 + cx^2 \ln x$	1293 $\ln y = a + bx^2 \ln x + \frac{c}{\ln x}$
1219 $\ln y = a + bx + c(\ln x)^2$	1244 $\ln y = a + bx \ln x + c \frac{\ln x}{x}$	1269 $\ln y = a + bx^2 + cx^2 \sqrt{x}$	1294 $\ln y = a + bx^2 \ln x + \frac{c}{\sqrt{x}}$
1220 $\ln y = a + bx + c \frac{x}{\ln x}$	1245 $\ln y = a + bx \ln x + \frac{c}{x}$	1270 $\ln y = a + bx^2 + cx^3$	1295 $\ln y = a + bx^2 \ln x + c \frac{\ln x}{x}$
1221 $\ln y = a + bx + c\sqrt{x}$	1246 $\ln y = a + bx \ln x + \frac{c}{x\sqrt{x}}$	1271 $\ln y = a + bx^2 + c e^x$	1296 $\ln y = a + bx^2 \ln x + \frac{c}{x}$

1297 $\ln y = a + bx^2 \ln x + \frac{c}{x\sqrt{x}}$	1322 $\ln y = a + bx^3 + \frac{c}{\ln x}$	1347 $\ln y = a + b\sqrt{x} \ln x + \frac{c}{\ln x}$	1372 $\ln y = a + b \frac{x}{\ln x} + \frac{c}{x\sqrt{x}}$
1298 $\ln y = a + bx^2 \ln x + c \frac{\ln x}{x^2}$	1323 $\ln y = a + bx^3 + \frac{c}{\sqrt{x}}$	1348 $\ln y = a + b\sqrt{x} \ln x + \frac{c}{\sqrt{x}}$	1373 $\ln y = a + b \frac{x}{\ln x} + c \frac{\ln x}{x^2}$
1299 $\ln y = a + bx^2 \ln x + \frac{c}{x^2}$	1324 $\ln y = a + bx^3 + c \frac{\ln x}{x}$	1349 $\ln y = a + b\sqrt{x} \ln x + c \frac{\ln x}{x}$	1374 $\ln y = a + b \frac{x}{\ln x} + \frac{c}{x^2}$
1300 $\ln y = a + bx^2 \ln x + c e^{-x}$	1325 $\ln y = a + bx^3 + \frac{c}{x}$	1350 $\ln y = a + b\sqrt{x} \ln x + \frac{c}{x}$	1375 $\ln y = a + b \frac{x}{\ln x} + c e^{-x}$
1301 $\ln y = a + bx^2 \sqrt{x} + cx^3$	1326 $\ln y = a + bx^3 + \frac{c}{x\sqrt{x}}$	1351 $\ln y = a + b\sqrt{x} \ln x + \frac{c}{x\sqrt{x}}$	1376 $\ln y = a + b\sqrt{x} + c \ln x$
1302 $\ln y = a + bx^2 \sqrt{x} + c e^x$	1327 $\ln y = a + bx^3 + c \frac{\ln x}{x^2}$	1352 $\ln y = a + b\sqrt{x} \ln x + c \frac{\ln x}{x^2}$	1377 $\ln y = a + b\sqrt{x} + \frac{c}{\ln x}$
1303 $\ln y = a + bx^2 \sqrt{x} + c\sqrt{x} \ln x$	1328 $\ln y = a + bx^3 + \frac{c}{x^2}$	1353 $\ln y = a + b\sqrt{x} \ln x + \frac{c}{x^2}$	1378 $\ln y = a + b\sqrt{x} + \frac{c}{\sqrt{x}}$
1304 $\ln y = a + bx^2 \sqrt{x} + c(\ln x)^2$	1329 $\ln y = a + bx^3 + c e^{-x}$	1354 $\ln y = a + b\sqrt{x} \ln x + c e^{-x}$	1379 $\ln y = a + b\sqrt{x} + c \frac{\ln x}{x}$
1305 $\ln y = a + bx^2 \sqrt{x} + c \frac{x}{\ln x}$	1330 $\ln y = a + b e^x + c\sqrt{x} \ln x$	1355 $\ln y = a + b(\ln x)^2 + c \frac{x}{\ln x}$	1380 $\ln y = a + b\sqrt{x} + \frac{c}{x}$
1306 $\ln y = a + bx^2 \sqrt{x} + c\sqrt{x}$	1331 $\ln y = a + b e^x + c(\ln x)^2$	1356 $\ln y = a + b(\ln x)^2 + c\sqrt{x}$	1381 $\ln y = a + b\sqrt{x} + \frac{c}{x\sqrt{x}}$
1307 $\ln y = a + bx^2 \sqrt{x} + c \ln x$	1332 $\ln y = a + b e^x + c \frac{x}{\ln x}$	1357 $\ln y = a + b(\ln x)^2 + c \ln x$	1382 $\ln y = a + b\sqrt{x} + c \frac{\ln x}{x^2}$
1308 $\ln y = a + bx^2 \sqrt{x} + \frac{c}{\ln x}$	1333 $\ln y = a + b e^x + c\sqrt{x}$	1358 $\ln y = a + b(\ln x)^2 + \frac{c}{\ln x}$	1383 $\ln y = a + b\sqrt{x} + \frac{c}{x^2}$
1309 $\ln y = a + bx^2 \sqrt{x} + \frac{c}{\sqrt{x}}$	1334 $\ln y = a + b e^x + c \ln x$	1359 $\ln y = a + b(\ln x)^2 + \frac{c}{\sqrt{x}}$	1384 $\ln y = a + b\sqrt{x} + c e^{-x}$
1310 $\ln y = a + bx^2 \sqrt{x} + c \frac{\ln x}{x}$	1335 $\ln y = a + b e^x + \frac{c}{\ln x}$	1360 $\ln y = a + b(\ln x)^2 + c \frac{\ln x}{x}$	1385 $\ln y = a + b \ln x + \frac{c}{\ln x}$
1311 $\ln y = a + bx^2 \sqrt{x} + \frac{c}{x}$	1336 $\ln y = a + b e^x + \frac{c}{\sqrt{x}}$	1361 $\ln y = a + b(\ln x)^2 + \frac{c}{x}$	1386 $\ln y = a + b \ln x + \frac{c}{\sqrt{x}}$
1312 $\ln y = a + bx^2 \sqrt{x} + \frac{c}{x\sqrt{x}}$	1337 $\ln y = a + b e^x + c \frac{\ln x}{x}$	1362 $\ln y = a + b(\ln x)^2 + \frac{c}{x\sqrt{x}}$	1387 $\ln y = a + b \ln x + c \frac{\ln x}{x}$
1313 $\ln y = a + bx^2 \sqrt{x} + c \frac{\ln x}{x^2}$	1338 $\ln y = a + b e^x + \frac{c}{x}$	1363 $\ln y = a + b(\ln x)^2 + c \frac{\ln x}{x^2}$	1388 $\ln y = a + b \ln x + \frac{c}{x}$
1314 $\ln y = a + bx^2 \sqrt{x} + \frac{c}{x^2}$	1339 $\ln y = a + b e^x + \frac{c}{x\sqrt{x}}$	1364 $\ln y = a + b(\ln x)^2 + \frac{c}{x^2}$	1389 $\ln y = a + b \ln x + \frac{c}{x\sqrt{x}}$
1315 $\ln y = a + bx^2 \sqrt{x} + c e^{-x}$	1340 $\ln y = a + b e^x + c \frac{\ln x}{x^2}$	1365 $\ln y = a + b(\ln x)^2 + c e^{-x}$	1390 $\ln y = a + b \ln x + c \frac{\ln x}{x^2}$
1316 $\ln y = a + bx^3 + c e^x$	1341 $\ln y = a + b e^x + \frac{c}{x^2}$	1366 $\ln y = a + b \frac{x}{\ln x} + c\sqrt{x}$	1391 $\ln y = a + b \ln x + \frac{c}{x^2}$
1317 $\ln y = a + bx^3 + c\sqrt{x} \ln x$	1342 $\ln y = a + b e^x + c e^{-x}$	1367 $\ln y = a + b \frac{x}{\ln x} + c \ln x$	1392 $\ln y = a + b \ln x + c e^{-x}$
1318 $\ln y = a + bx^3 + c(\ln x)^2$	1343 $\ln y = a + b\sqrt{x} \ln x + c(\ln x)^2$	1368 $\ln y = a + b \frac{x}{\ln x} + \frac{c}{\ln x}$	1393 $\ln y = a + \frac{b}{\ln x} + \frac{c}{\sqrt{x}}$
1319 $\ln y = a + bx^3 + c \frac{x}{\ln x}$	1344 $\ln y = a + b\sqrt{x} \ln x + c \frac{x}{\ln x}$	1369 $\ln y = a + b \frac{x}{\ln x} + \frac{c}{\sqrt{x}}$	1394 $\ln y = a + \frac{b}{\ln x} + c \frac{\ln x}{x}$
1320 $\ln y = a + bx^3 + c\sqrt{x}$	1345 $\ln y = a + b\sqrt{x} \ln x + c\sqrt{x}$	1370 $\ln y = a + b \frac{x}{\ln x} + c \frac{\ln x}{x}$	1395 $\ln y = a + \frac{b}{\ln x} + \frac{c}{x}$
1321 $\ln y = a + bx^3 + c \ln x$	1346 $\ln y = a + b\sqrt{x} \ln x + c \ln x$	1371 $\ln y = a + b \frac{x}{\ln x} + \frac{c}{x}$	1396 $\ln y = a + \frac{b}{\ln x} + \frac{c}{x\sqrt{x}}$

1397 $\ln y = a + \frac{b}{\ln x} + c \frac{\ln x}{x^2}$	1422 $\sqrt[y]{y} = a + bx + cx\sqrt{x}$	1447 $\sqrt[y]{y} = a + bx \ln x + c\sqrt{x} \ln x$	1472 $\sqrt[y]{y} = a + bx\sqrt{x} + c \frac{\ln x}{x}$
1398 $\ln y = a + \frac{b}{\ln x} + \frac{c}{x^2}$	1423 $\sqrt[y]{y} = a + bx + cx^2$	1448 $\sqrt[y]{y} = a + bx \ln x + c(\ln x)^2$	1473 $\sqrt[y]{y} = a + bx\sqrt{x} + \frac{c}{x}$
1399 $\ln y = a + \frac{b}{\ln x} + c e^{-x}$	1424 $\sqrt[y]{y} = a + bx + cx^2 \ln x$	1449 $\sqrt[y]{y} = a + bx \ln x + c \frac{x}{\ln x}$	1474 $\sqrt[y]{y} = a + bx\sqrt{x} + \frac{c}{x\sqrt{x}}$
1400 $\ln y = a + \frac{b}{\sqrt{x}} + c \frac{\ln x}{x}$	1425 $\sqrt[y]{y} = a + bx + cx^2 \sqrt{x}$	1450 $\sqrt[y]{y} = a + bx \ln x + c\sqrt{x}$	1475 $\sqrt[y]{y} = a + bx\sqrt{x} + c \frac{\ln x}{x^2}$
1401 $\ln y = a + \frac{b}{\sqrt{x}} + \frac{c}{x}$	1426 $\sqrt[y]{y} = a + bx + cx^3$	1451 $\sqrt[y]{y} = a + bx \ln x + c \ln x$	1476 $\sqrt[y]{y} = a + bx\sqrt{x} + \frac{c}{x^2}$
1402 $\ln y = a + \frac{b}{\sqrt{x}} + \frac{c}{x\sqrt{x}}$	1427 $\sqrt[y]{y} = a + bx + c e^x$	1452 $\sqrt[y]{y} = a + bx \ln x + \frac{c}{\ln x}$	1477 $\sqrt[y]{y} = a + bx\sqrt{x} + c e^{-x}$
1403 $\ln y = a + \frac{b}{\sqrt{x}} + c \frac{\ln x}{x^2}$	1428 $\sqrt[y]{y} = a + bx + c\sqrt{x} \ln x$	1453 $\sqrt[y]{y} = a + bx \ln x + \frac{c}{\sqrt{x}}$	1478 $\sqrt[y]{y} = a + bx^2 + cx^2 \ln x$
1404 $\ln y = a + \frac{b}{\sqrt{x}} + \frac{c}{x^2}$	1429 $\sqrt[y]{y} = a + bx + c(\ln x)^2$	1454 $\sqrt[y]{y} = a + bx \ln x + c \frac{\ln x}{x}$	1479 $\sqrt[y]{y} = a + bx^2 + cx^2 \sqrt{x}$
1405 $\ln y = a + \frac{b}{\sqrt{x}} + c e^{-x}$	1430 $\sqrt[y]{y} = a + bx + c \frac{x}{\ln x}$	1455 $\sqrt[y]{y} = a + bx \ln x + \frac{c}{x}$	1480 $\sqrt[y]{y} = a + bx^2 + cx^3$
1406 $\ln y = a + b \frac{\ln x}{x} + \frac{c}{x}$	1431 $\sqrt[y]{y} = a + bx + c\sqrt{x}$	1456 $\sqrt[y]{y} = a + bx \ln x + \frac{c}{x\sqrt{x}}$	1481 $\sqrt[y]{y} = a + bx^2 + c e^x$
1407 $\ln y = a + b \frac{\ln x}{x} + \frac{c}{x\sqrt{x}}$	1432 $\sqrt[y]{y} = a + bx + c \ln x$	1457 $\sqrt[y]{y} = a + bx \ln x + c \frac{\ln x}{x^2}$	1482 $\sqrt[y]{y} = a + bx^2 + c\sqrt{x} \ln x$
1408 $\ln y = a + b \frac{\ln x}{x} + c \frac{\ln x}{x^2}$	1433 $\sqrt[y]{y} = a + bx + \frac{c}{\ln x}$	1458 $\sqrt[y]{y} = a + bx \ln x + \frac{c}{x^2}$	1483 $\sqrt[y]{y} = a + bx^2 + c(\ln x)^2$
1409 $\ln y = a + b \frac{\ln x}{x} + \frac{c}{x^2}$	1434 $\sqrt[y]{y} = a + bx + \frac{c}{\sqrt{x}}$	1459 $\sqrt[y]{y} = a + bx \ln x + c e^{-x}$	1484 $\sqrt[y]{y} = a + bx^2 + c \frac{x}{\ln x}$
1410 $\ln y = a + b \frac{\ln x}{x} + c e^{-x}$	1435 $\sqrt[y]{y} = a + bx + c \frac{\ln x}{x}$	1460 $\sqrt[y]{y} = a + bx\sqrt{x} + cx^2$	1485 $\sqrt[y]{y} = a + bx^2 + c\sqrt{x}$
1411 $\ln y = a + \frac{b}{x} + \frac{c}{x\sqrt{x}}$	1436 $\sqrt[y]{y} = a + bx + \frac{c}{x}$	1461 $\sqrt[y]{y} = a + bx\sqrt{x} + cx^2 \ln x$	1486 $\sqrt[y]{y} = a + bx^2 + c \ln x$
1412 $\ln y = a + \frac{b}{x} + c \frac{\ln x}{x^2}$	1437 $\sqrt[y]{y} = a + bx + \frac{c}{x\sqrt{x}}$	1462 $\sqrt[y]{y} = a + bx\sqrt{x} + cx^2 \sqrt{x}$	1487 $\sqrt[y]{y} = a + bx^2 + \frac{c}{\ln x}$
1413 $\ln y = a + \frac{b}{x} + \frac{c}{x^2}$	1438 $\sqrt[y]{y} = a + bx + c \frac{\ln x}{x^2}$	1463 $\sqrt[y]{y} = a + bx\sqrt{x} + cx^3$	1488 $\sqrt[y]{y} = a + bx^2 + \frac{c}{\sqrt{x}}$
1414 $\ln y = a + \frac{b}{x} + c e^{-x}$	1439 $\sqrt[y]{y} = a + bx + \frac{c}{x^2}$	1464 $\sqrt[y]{y} = a + bx\sqrt{x} + c e^x$	1489 $\sqrt[y]{y} = a + bx^2 + c \frac{\ln x}{x}$
1415 $\ln y = a + \frac{b}{x\sqrt{x}} + c \frac{\ln x}{x^2}$	1440 $\sqrt[y]{y} = a + bx + c e^{-x}$	1465 $\sqrt[y]{y} = a + bx\sqrt{x} + c\sqrt{x} \ln x$	1490 $\sqrt[y]{y} = a + bx^2 + \frac{c}{x}$
1416 $\ln y = a + \frac{b}{x\sqrt{x}} + \frac{c}{x^2}$	1441 $\sqrt[y]{y} = a + bx \ln x + cx\sqrt{x}$	1466 $\sqrt[y]{y} = a + bx\sqrt{x} + c(\ln x)^2$	1491 $\sqrt[y]{y} = a + bx^2 + \frac{c}{x\sqrt{x}}$
1417 $\ln y = a + \frac{b}{x\sqrt{x}} + c e^{-x}$	1442 $\sqrt[y]{y} = a + bx \ln x + cx^2$	1467 $\sqrt[y]{y} = a + bx\sqrt{x} + c \frac{x}{\ln x}$	1492 $\sqrt[y]{y} = a + bx^2 + c \frac{\ln x}{x^2}$
1418 $\ln y = a + b \frac{\ln x}{x^2} + \frac{c}{x^2}$	1443 $\sqrt[y]{y} = a + bx \ln x + cx^2 \ln x$	1468 $\sqrt[y]{y} = a + bx\sqrt{x} + c\sqrt{x}$	1493 $\sqrt[y]{y} = a + bx^2 + \frac{c}{x^2}$
1419 $\ln y = a + b \frac{\ln x}{x^2} + c e^{-x}$	1444 $\sqrt[y]{y} = a + bx \ln x + cx^2 \sqrt{x}$	1469 $\sqrt[y]{y} = a + bx\sqrt{x} + c \ln x$	1494 $\sqrt[y]{y} = a + bx^2 + c e^{-x}$
1420 $\ln y = a + \frac{b}{x^2} + c e^{-x}$	1445 $\sqrt[y]{y} = a + bx \ln x + cx^3$	1470 $\sqrt[y]{y} = a + bx\sqrt{x} + \frac{c}{\ln x}$	1495 $\sqrt[y]{y} = a + bx^2 \ln x + cx^2 \sqrt{x}$
1421 $\sqrt[y]{y} = a + bx + cx \ln x$	1446 $\sqrt[y]{y} = a + bx \ln x + c e^x$	1471 $\sqrt[y]{y} = a + bx\sqrt{x} + \frac{c}{\sqrt{x}}$	1496 $\sqrt[y]{y} = a + bx^2 \ln x + cx^3$

1497 $y = a + bx^2 \ln x + ce^x$	1522 $y = a + bx^2 \sqrt{x} + \frac{c}{x\sqrt{x}}$	1547 $y = a + be^x + c \frac{\ln x}{x}$	1572 $y = a + b(\ln x)^2 + \frac{c}{x\sqrt{x}}$
1498 $y = a + bx^2 \ln x + c\sqrt{x} \ln x$	1523 $y = a + bx^2 \sqrt{x} + c \frac{\ln x}{x^2}$	1548 $y = a + be^x + \frac{c}{x}$	1573 $y = a + b(\ln x)^2 + c \frac{\ln x}{x^2}$
1499 $y = a + bx^2 \ln x + c(\ln x)^2$	1524 $y = a + bx^2 \sqrt{x} + \frac{c}{x^2}$	1549 $y = a + be^x + \frac{c}{x\sqrt{x}}$	1574 $y = a + b(\ln x)^2 + \frac{c}{x^2}$
1500 $y = a + bx^2 \ln x + c \frac{x}{\ln x}$	1525 $y = a + bx^2 \sqrt{x} + ce^{-x}$	1550 $y = a + be^x + c \frac{\ln x}{x^2}$	1575 $y = a + b(\ln x)^2 + ce^{-x}$
1501 $y = a + bx^2 \ln x + c\sqrt{x}$	1526 $y = a + bx^3 + ce^x$	1551 $y = a + be^x + \frac{c}{x^2}$	1576 $y = a + b \frac{x}{\ln x} + c\sqrt{x}$
1502 $y = a + bx^2 \ln x + c \ln x$	1527 $y = a + bx^3 + c\sqrt{x} \ln x$	1552 $y = a + be^x + ce^{-x}$	1577 $y = a + b \frac{x}{\ln x} + c \ln x$
1503 $y = a + bx^2 \ln x + \frac{c}{\ln x}$	1528 $y = a + bx^3 + c(\ln x)^2$	1553 $y = a + b\sqrt{x} \ln x + c(\ln x)^2$	1578 $y = a + b \frac{x}{\ln x} + \frac{c}{\ln x}$
1504 $y = a + bx^2 \ln x + \frac{c}{\sqrt{x}}$	1529 $y = a + bx^3 + c \frac{x}{\ln x}$	1554 $y = a + b\sqrt{x} \ln x + c \frac{x}{\ln x}$	1579 $y = a + b \frac{x}{\ln x} + \frac{c}{\sqrt{x}}$
1505 $y = a + bx^2 \ln x + c \frac{\ln x}{x}$	1530 $y = a + bx^3 + c\sqrt{x}$	1555 $y = a + b\sqrt{x} \ln x + c\sqrt{x}$	1580 $y = a + b \frac{x}{\ln x} + c \frac{\ln x}{x}$
1506 $y = a + bx^2 \ln x + \frac{c}{x}$	1531 $y = a + bx^3 + c \ln x$	1556 $y = a + b\sqrt{x} \ln x + c \ln x$	1581 $y = a + b \frac{x}{\ln x} + \frac{c}{x}$
1507 $y = a + bx^2 \ln x + \frac{c}{x\sqrt{x}}$	1532 $y = a + bx^3 + \frac{c}{\ln x}$	1557 $y = a + b\sqrt{x} \ln x + \frac{c}{\ln x}$	1582 $y = a + b \frac{x}{\ln x} + \frac{c}{x\sqrt{x}}$
1508 $y = a + bx^2 \ln x + c \frac{\ln x}{x^2}$	1533 $y = a + bx^3 + \frac{c}{\sqrt{x}}$	1558 $y = a + b\sqrt{x} \ln x + \frac{c}{\sqrt{x}}$	1583 $y = a + b \frac{x}{\ln x} + c \frac{\ln x}{x^2}$
1509 $y = a + bx^2 \ln x + \frac{c}{x^2}$	1534 $y = a + bx^3 + c \frac{\ln x}{x}$	1559 $y = a + b\sqrt{x} \ln x + c \frac{\ln x}{x}$	1584 $y = a + b \frac{x}{\ln x} + \frac{c}{x^2}$
1510 $y = a + bx^2 \ln x + ce^{-x}$	1535 $y = a + bx^3 + \frac{c}{x}$	1560 $y = a + b\sqrt{x} \ln x + \frac{c}{x}$	1585 $y = a + b \frac{x}{\ln x} + ce^{-x}$
1511 $y = a + bx^2 \sqrt{x} + cx^3$	1536 $y = a + bx^3 + \frac{c}{x\sqrt{x}}$	1561 $y = a + b\sqrt{x} \ln x + \frac{c}{x\sqrt{x}}$	1586 $y = a + b\sqrt{x} + c \ln x$
1512 $y = a + bx^2 \sqrt{x} + ce^x$	1537 $y = a + bx^3 + c \frac{\ln x}{x^2}$	1562 $y = a + b\sqrt{x} \ln x + c \frac{\ln x}{x^2}$	1587 $y = a + b\sqrt{x} + \frac{c}{\ln x}$
1513 $y = a + bx^2 \sqrt{x} + c\sqrt{x} \ln x$	1538 $y = a + bx^3 + \frac{c}{x^2}$	1563 $y = a + b\sqrt{x} \ln x + \frac{c}{x^2}$	1588 $y = a + b\sqrt{x} + \frac{c}{\sqrt{x}}$
1514 $y = a + bx^2 \sqrt{x} + c(\ln x)^2$	1539 $y = a + bx^3 + ce^{-x}$	1564 $y = a + b\sqrt{x} \ln x + ce^{-x}$	1589 $y = a + b\sqrt{x} + c \frac{\ln x}{x}$
1515 $y = a + bx^2 \sqrt{x} + c \frac{x}{\ln x}$	1540 $y = a + be^x + c\sqrt{x} \ln x$	1565 $y = a + b(\ln x)^2 + c \frac{x}{\ln x}$	1590 $y = a + b\sqrt{x} + \frac{c}{x}$
1516 $y = a + bx^2 \sqrt{x} + c\sqrt{x}$	1541 $y = a + be^x + c(\ln x)^2$	1566 $y = a + b(\ln x)^2 + c\sqrt{x}$	1591 $y = a + b\sqrt{x} + \frac{c}{x\sqrt{x}}$
1517 $y = a + bx^2 \sqrt{x} + c \ln x$	1542 $y = a + be^x + c \frac{x}{\ln x}$	1567 $y = a + b(\ln x)^2 + c \ln x$	1592 $y = a + b\sqrt{x} + c \frac{\ln x}{x^2}$
1518 $y = a + bx^2 \sqrt{x} + \frac{c}{\ln x}$	1543 $y = a + be^x + c\sqrt{x}$	1568 $y = a + b(\ln x)^2 + \frac{c}{\ln x}$	1593 $y = a + b\sqrt{x} + \frac{c}{x^2}$
1519 $y = a + bx^2 \sqrt{x} + \frac{c}{\sqrt{x}}$	1544 $y = a + be^x + c \ln x$	1569 $y = a + b(\ln x)^2 + \frac{c}{\sqrt{x}}$	1594 $y = a + b\sqrt{x} + ce^{-x}$
1520 $y = a + bx^2 \sqrt{x} + c \frac{\ln x}{x}$	1545 $y = a + be^x + \frac{c}{\ln x}$	1570 $y = a + b(\ln x)^2 + c \frac{\ln x}{x}$	1595 $y = a + b \ln x + \frac{c}{\ln x}$
1521 $y = a + bx^2 \sqrt{x} + \frac{c}{x}$	1546 $y = a + be^x + \frac{c}{\sqrt{x}}$	1571 $y = a + b(\ln x)^2 + \frac{c}{x}$	1596 $y = a + b \ln x + \frac{c}{\sqrt{x}}$

$$1597 \quad y = a + b \ln x + c \frac{\ln x}{x}$$

$$1598 \quad y = a + b \ln x + \frac{c}{x}$$

$$1599 \quad y = a + b \ln x + \frac{c}{x\sqrt{x}}$$

$$1600 \quad y = a + b \ln x + c \frac{\ln x}{x^2}$$

$$1601 \quad y = a + b \ln x + \frac{c}{x^2}$$

$$1602 \quad y = a + b \ln x + c e^{-x}$$

$$1603 \quad y = a + \frac{b}{\ln x} + \frac{c}{\sqrt{x}}$$

$$1604 \quad y = a + \frac{b}{\ln x} + c \frac{\ln x}{x}$$

$$1605 \quad y = a + \frac{b}{\ln x} + \frac{c}{x}$$

$$1606 \quad y = a + \frac{b}{\ln x} + \frac{c}{x\sqrt{x}}$$

$$1607 \quad y = a + \frac{b}{\ln x} + c \frac{\ln x}{x^2}$$

$$1608 \quad y = a + \frac{b}{\ln x} + \frac{c}{x^2}$$

$$1609 \quad y = a + \frac{b}{\ln x} + c e^{-x}$$

$$1610 \quad y = a + \frac{b}{\sqrt{x}} + c \frac{\ln x}{x}$$

$$1611 \quad y = a + \frac{b}{\sqrt{x}} + \frac{c}{x}$$

$$1612 \quad y = a + \frac{b}{\sqrt{x}} + \frac{c}{x\sqrt{x}}$$

$$1613 \quad y = a + \frac{b}{\sqrt{x}} + c \frac{\ln x}{x^2}$$

$$1614 \quad y = a + \frac{b}{\sqrt{x}} + \frac{c}{x^2}$$

$$1615 \quad y = a + \frac{b}{\sqrt{x}} + c e^{-x}$$

$$1616 \quad y = a + b \frac{\ln x}{x} + \frac{c}{x}$$

$$1617 \quad y = a + b \frac{\ln x}{x} + \frac{c}{x\sqrt{x}}$$

$$1618 \quad y = a + b \frac{\ln x}{x} + c \frac{\ln x}{x^2}$$

$$1619 \quad y = a + b \frac{\ln x}{x} + \frac{c}{x^2}$$

$$1620 \quad y = a + b \frac{\ln x}{x} + c e^{-x}$$

$$1621 \quad y = a + \frac{b}{x} + \frac{c}{x\sqrt{x}}$$

$$1622 \quad y = a + \frac{b}{x} + c \frac{\ln x}{x^2}$$

$$1623 \quad y = a + \frac{b}{x} + \frac{c}{x^2}$$

$$1624 \quad y = a + \frac{b}{x} + c e^{-x}$$

$$1625 \quad y = a + \frac{b}{x\sqrt{x}} + c \frac{\ln x}{x^2}$$

$$1626 \quad y = a + \frac{b}{x\sqrt{x}} + \frac{c}{x^2}$$

$$1627 \quad y = a + \frac{b}{x\sqrt{x}} + c e^{-x}$$

$$1628 \quad y = a + b \frac{\ln x}{x^2} + \frac{c}{x^2}$$

$$1629 \quad y = a + b \frac{\ln x}{x^2} + c e^{-x}$$

$$1630 \quad y = a + \frac{b}{x^2} + c e^{-x}$$

Eqns 2001-3330: Three Term Functions with Intercept

$$2001 \ y = a + bx + cx \ln x + dx\sqrt{x}$$

$$2002 \ y = a + bx + cx \ln x + dx^2$$

$$2003 \ y = a + bx + cx \ln x + dx^2 \ln x$$

$$2004 \ y = a + bx + cx \ln x + dx^2 \sqrt{x}$$

$$2005 \ y = a + bx + cx \ln x + dx^3$$

$$2006 \ y = a + bx + cx \ln x + d e^x$$

$$2007 \ y = a + bx + cx \ln x + d\sqrt{x} \ln x$$

$$2008 \ y = a + bx + cx \ln x + d(\ln x)^2$$

$$2009 \ y = a + bx + cx \ln x + d \frac{x}{\ln x}$$

$$2010 \ y = a + bx + cx \ln x + d\sqrt{x}$$

$$2011 \ y = a + bx + cx \ln x + d \ln x$$

$$2012 \ y = a + bx + cx \ln x + \frac{d}{\ln x}$$

$$2013 \ y = a + bx + cx \ln x + \frac{d}{\sqrt{x}}$$

$$2014 \ y = a + bx + cx \ln x + d \frac{\ln x}{x}$$

$$2015 \ y = a + bx + cx \ln x + \frac{d}{x}$$

$$2016 \ y = a + bx + cx \ln x + \frac{d}{x\sqrt{x}}$$

$$2017 \ y = a + bx + cx \ln x + d \frac{\ln x}{x^2}$$

$$2018 \ y = a + bx + cx \ln x + \frac{d}{x^2}$$

$$2019 \ y = a + bx + cx \ln x + d e^{-x}$$

$$2020 \ y = a + bx + cx\sqrt{x} + dx^2$$

$$2021 \ y = a + bx + cx\sqrt{x} + dx^2 \ln x$$

$$2022 \ y = a + bx + cx\sqrt{x} + dx^2 \sqrt{x}$$

$$2023 \ y = a + bx + cx\sqrt{x} + dx^3$$

$$2024 \ y = a + bx + cx\sqrt{x} + d e^x$$

$$2025 \ y = a + bx + cx\sqrt{x} + d\sqrt{x} \ln x$$

$$2026 \ y = a + bx + cx\sqrt{x} + d(\ln x)^2$$

$$2027 \ y = a + bx + cx\sqrt{x} + d \frac{x}{\ln x}$$

$$2028 \ y = a + bx + cx\sqrt{x} + d\sqrt{x}$$

$$2029 \ y = a + bx + cx\sqrt{x} + d \ln x$$

$$2030 \ y = a + bx + cx\sqrt{x} + \frac{d}{\ln x}$$

$$2031 \ y = a + bx + cx\sqrt{x} + \frac{d}{\sqrt{x}}$$

$$2032 \ y = a + bx + cx\sqrt{x} + d \frac{\ln x}{x}$$

$$2033 \ y = a + bx + cx\sqrt{x} + \frac{d}{x}$$

$$2034 \ y = a + bx + cx\sqrt{x} + \frac{d}{x\sqrt{x}}$$

$$2035 \ y = a + bx + cx\sqrt{x} + d \frac{\ln x}{x^2}$$

$$2036 \ y = a + bx + cx\sqrt{x} + \frac{d}{x^2}$$

$$2037 \ y = a + bx + cx\sqrt{x} + d e^{-x}$$

$$2038 \ y = a + bx + cx^2 + dx^2 \ln x$$

$$2039 \ y = a + bx + cx^2 + dx^2 \sqrt{x}$$

$$2040 \ y = a + bx + cx^2 + dx^3$$

$$2041 \ y = a + bx + cx^2 + d e^x$$

$$2042 \ y = a + bx + cx^2 + d\sqrt{x} \ln x$$

$$2043 \ y = a + bx + cx^2 + d(\ln x)^2$$

$$2044 \ y = a + bx + cx^2 + d \frac{x}{\ln x}$$

$$2045 \ y = a + bx + cx^2 + d\sqrt{x}$$

$$2046 \ y = a + bx + cx^2 + d \ln x$$

$$2047 \ y = a + bx + cx^2 + \frac{d}{\ln x}$$

$$2048 \ y = a + bx + cx^2 + \frac{d}{\sqrt{x}}$$

$$2049 \ y = a + bx + cx^2 + d \frac{\ln x}{x}$$

$$2050 \ y = a + bx + cx^2 + \frac{d}{x}$$

$$2051 \ y = a + bx + cx^2 + \frac{d}{x\sqrt{x}}$$

$$2052 \ y = a + bx + cx^2 + d \frac{\ln x}{x^2}$$

$$2053 \ y = a + bx + cx^2 + \frac{d}{x^2}$$

$$2054 \ y = a + bx + cx^2 + d e^{-x}$$

$$2055 \ y = a + bx + cx^2 \ln x + dx^2 \sqrt{x}$$

$$2056 \ y = a + bx + cx^2 \ln x + dx^3$$

$$2057 \ y = a + bx + cx^2 \ln x + d e^x$$

$$2058 \ y = a + bx + cx^2 \ln x + d\sqrt{x} \ln x$$

$$2059 \ y = a + bx + cx^2 \ln x + d(\ln x)^2$$

$$2060 \ y = a + bx + cx^2 \ln x + d \frac{x}{\ln x}$$

$$2061 \ y = a + bx + cx^2 \ln x + d\sqrt{x}$$

$$2062 \ y = a + bx + cx^2 \ln x + d \ln x$$

$$2063 \ y = a + bx + cx^2 \ln x + \frac{d}{\ln x}$$

$$2064 \ y = a + bx + cx^2 \ln x + \frac{d}{\sqrt{x}}$$

$$2065 \ y = a + bx + cx^2 \ln x + d \frac{\ln x}{x}$$

$$2066 \ y = a + bx + cx^2 \ln x + \frac{d}{x}$$

$$2067 \ y = a + bx + cx^2 \ln x + \frac{d}{x\sqrt{x}}$$

$$2068 \ y = a + bx + cx^2 \ln x + d \frac{\ln x}{x^2}$$

$$2069 \ y = a + bx + cx^2 \ln x + \frac{d}{x^2}$$

$$2070 \ y = a + bx + cx^2 \ln x + d e^{-x}$$

$$2071 \ y = a + bx + cx^2 \sqrt{x} + dx^3$$

$$2072 \ y = a + bx + cx^2 \sqrt{x} + d e^x$$

2073 $y = a + bx + cx^2 \sqrt{x} + d\sqrt{x} \ln x$

2074 $y = a + bx + cx^2 \sqrt{x} + d(\ln x)^2$

2075 $y = a + bx + cx^2 \sqrt{x} + d \frac{x}{\ln x}$

2076 $y = a + bx + cx^2 \sqrt{x} + d\sqrt{x}$

2077 $y = a + bx + cx^2 \sqrt{x} + d \ln x$

2078 $y = a + bx + cx^2 \sqrt{x} + \frac{d}{\ln x}$

2079 $y = a + bx + cx^2 \sqrt{x} + \frac{d}{\sqrt{x}}$

2080 $y = a + bx + cx^2 \sqrt{x} + d \frac{\ln x}{x}$

2081 $y = a + bx + cx^2 \sqrt{x} + \frac{d}{x}$

2082 $y = a + bx + cx^2 \sqrt{x} + \frac{d}{x\sqrt{x}}$

2083 $y = a + bx + cx^2 \sqrt{x} + d \frac{\ln x}{x^2}$

2084 $y = a + bx + cx^2 \sqrt{x} + \frac{d}{x^2}$

2085 $y = a + bx + cx^2 \sqrt{x} + d e^{-x}$

2086 $y = a + bx + cx^3 + d e^x$

2087 $y = a + bx + cx^3 + d\sqrt{x} \ln x$

2088 $y = a + bx + cx^3 + d(\ln x)^2$

2089 $y = a + bx + cx^3 + d \frac{x}{\ln x}$

2090 $y = a + bx + cx^3 + d\sqrt{x}$

2091 $y = a + bx + cx^3 + d \ln x$

2092 $y = a + bx + cx^3 + \frac{d}{\ln x}$

2093 $y = a + bx + cx^3 + \frac{d}{\sqrt{x}}$

2094 $y = a + bx + cx^3 + d \frac{\ln x}{x}$

2095 $y = a + bx + cx^3 + \frac{d}{x}$

2096 $y = a + bx + cx^3 + \frac{d}{x\sqrt{x}}$

2097 $y = a + bx + cx^3 + d \frac{\ln x}{x^2}$

2098 $y = a + bx + cx^3 + \frac{d}{x^2}$

2099 $y = a + bx + cx^3 + d e^{-x}$

2100 $y = a + bx + c e^x + d\sqrt{x} \ln x$

2101 $y = a + bx + c e^x + d(\ln x)^2$

2102 $y = a + bx + c e^x + d \frac{x}{\ln x}$

2103 $y = a + bx + c e^x + d\sqrt{x}$

2104 $y = a + bx + c e^x + d \ln x$

2105 $y = a + bx + c e^x + \frac{d}{\ln x}$

2106 $y = a + bx + c e^x + \frac{d}{\sqrt{x}}$

2107 $y = a + bx + c e^x + d \frac{\ln x}{x}$

2108 $y = a + bx + c e^x + \frac{d}{x}$

2109 $y = a + bx + c e^x + \frac{d}{x\sqrt{x}}$

2110 $y = a + bx + c e^x + d \frac{\ln x}{x^2}$

2111 $y = a + bx + c e^x + \frac{d}{x^2}$

2112 $y = a + bx + c e^x + d e^{-x}$

2113 $y = a + bx + c\sqrt{x} \ln x + d(\ln x)^2$

2114 $y = a + bx + c\sqrt{x} \ln x + d \frac{x}{\ln x}$

2115 $y = a + bx + c\sqrt{x} \ln x + d\sqrt{x}$

2116 $y = a + bx + c\sqrt{x} \ln x + d \ln x$

2117 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{\ln x}$

2118 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{\sqrt{x}}$

2119 $y = a + bx + c\sqrt{x} \ln x + d \frac{\ln x}{x}$

2120 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{x}$

2121 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{x\sqrt{x}}$

2122 $y = a + bx + c\sqrt{x} \ln x + d \frac{\ln x}{x^2}$

2123 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{x^2}$

2124 $y = a + bx + c\sqrt{x} \ln x + d e^{-x}$

2125 $y = a + bx + c(\ln x)^2 + d \frac{x}{\ln x}$

2126 $y = a + bx + c(\ln x)^2 + d\sqrt{x}$

2127 $y = a + bx + c(\ln x)^2 + d \ln x$

2128 $y = a + bx + c(\ln x)^2 + \frac{d}{\ln x}$

2129 $y = a + bx + c(\ln x)^2 + \frac{d}{\sqrt{x}}$

2130 $y = a + bx + c(\ln x)^2 + d \frac{\ln x}{x}$

2131 $y = a + bx + c(\ln x)^2 + \frac{d}{x}$

2132 $y = a + bx + c(\ln x)^2 + \frac{d}{x\sqrt{x}}$

2133 $y = a + bx + c(\ln x)^2 + d \frac{\ln x}{x^2}$

2134 $y = a + bx + c(\ln x)^2 + \frac{d}{x^2}$

2135 $y = a + bx + c(\ln x)^2 + d e^{-x}$

2136 $y = a + bx + c \frac{x}{\ln x} + d\sqrt{x}$

2137 $y = a + bx + c \frac{x}{\ln x} + d \ln x$

2138 $y = a + bx + c \frac{x}{\ln x} + \frac{d}{\ln x}$

2139 $y = a + bx + c \frac{x}{\ln x} + \frac{d}{\sqrt{x}}$

2140 $y = a + bx + c \frac{x}{\ln x} + d \frac{\ln x}{x}$

2141 $y = a + bx + c \frac{x}{\ln x} + \frac{d}{x}$

2142 $y = a + bx + c \frac{x}{\ln x} + \frac{d}{x\sqrt{x}}$

2143 $y = a + bx + c \frac{x}{\ln x} + d \frac{\ln x}{x^2}$

2144 $y = a + bx + c \frac{x}{\ln x} + \frac{d}{x^2}$

2145 $y = a + bx + c \frac{x}{\ln x} + d e^{-x}$

2146 $y = a + bx + c\sqrt{x} + d \ln x$

2147 $y = a + bx + c\sqrt{x} + \frac{d}{\ln x}$

2148 $y = a + bx + c\sqrt{x} + \frac{d}{\sqrt{x}}$

2149 $y = a + bx + c\sqrt{x} + d \frac{\ln x}{x}$

2150 $y = a + bx + c\sqrt{x} + \frac{d}{x}$

- 2151 $y = a + bx + c\sqrt{x} + \frac{d}{x\sqrt{x}}$
- 2152 $y = a + bx + c\sqrt{x} + d\frac{\ln x}{x^2}$
- 2153 $y = a + bx + c\sqrt{x} + \frac{d}{x^2}$
- 2154 $y = a + bx + c\sqrt{x} + d e^{-x}$
- 2155 $y = a + bx + c \ln x + \frac{d}{\ln x}$
- 2156 $y = a + bx + c \ln x + \frac{d}{\sqrt{x}}$
- 2157 $y = a + bx + c \ln x + d\frac{\ln x}{x}$
- 2158 $y = a + bx + c \ln x + \frac{d}{x}$
- 2159 $y = a + bx + c \ln x + \frac{d}{x\sqrt{x}}$
- 2160 $y = a + bx + c \ln x + d\frac{\ln x}{x^2}$
- 2161 $y = a + bx + c \ln x + \frac{d}{x^2}$
- 2162 $y = a + bx + c \ln x + d e^{-x}$
- 2163 $y = a + bx + \frac{c}{\ln x} + \frac{d}{\sqrt{x}}$
- 2164 $y = a + bx + \frac{c}{\ln x} + d\frac{\ln x}{x}$
- 2165 $y = a + bx + \frac{c}{\ln x} + \frac{d}{x}$
- 2166 $y = a + bx + \frac{c}{\ln x} + \frac{d}{x\sqrt{x}}$
- 2167 $y = a + bx + \frac{c}{\ln x} + d\frac{\ln x}{x^2}$
- 2168 $y = a + bx + \frac{c}{\ln x} + \frac{d}{x^2}$
- 2169 $y = a + bx + \frac{c}{\ln x} + d e^{-x}$
- 2170 $y = a + bx + \frac{c}{\sqrt{x}} + d\frac{\ln x}{x}$
- 2171 $y = a + bx + \frac{c}{\sqrt{x}} + \frac{d}{x}$
- 2172 $y = a + bx + \frac{c}{\sqrt{x}} + \frac{d}{x\sqrt{x}}$
- 2173 $y = a + bx + \frac{c}{\sqrt{x}} + d\frac{\ln x}{x^2}$
- 2174 $y = a + bx + \frac{c}{\sqrt{x}} + \frac{d}{x^2}$
- 2175 $y = a + bx + \frac{c}{\sqrt{x}} + d e^{-x}$
- 2176 $y = a + bx + c\frac{\ln x}{x} + \frac{d}{x}$
- 2177 $y = a + bx + c\frac{\ln x}{x} + \frac{d}{x\sqrt{x}}$
- 2178 $y = a + bx + c\frac{\ln x}{x} + d\frac{\ln x}{x^2}$
- 2179 $y = a + bx + c\frac{\ln x}{x} + \frac{d}{x^2}$
- 2180 $y = a + bx + c\frac{\ln x}{x} + d e^{-x}$
- 2181 $y = a + bx + \frac{c}{x} + \frac{d}{x\sqrt{x}}$
- 2182 $y = a + bx + \frac{c}{x} + d\frac{\ln x}{x^2}$
- 2183 $y = a + bx + \frac{c}{x} + \frac{d}{x^2}$
- 2184 $y = a + bx + \frac{c}{x} + d e^{-x}$
- 2185 $y = a + bx + \frac{c}{x\sqrt{x}} + d\frac{\ln x}{x^2}$
- 2186 $y = a + bx + \frac{c}{x\sqrt{x}} + \frac{d}{x^2}$
- 2187 $y = a + bx + \frac{c}{x\sqrt{x}} + d e^{-x}$
- 2188 $y = a + bx + c\frac{\ln x}{x^2} + \frac{d}{x^2}$
- 2189 $y = a + bx + c\frac{\ln x}{x^2} + d e^{-x}$
- 2190 $y = a + bx + \frac{c}{x^2} + d e^{-x}$
- 2191 $y = a + bx \ln x + cx\sqrt{x} + dx^2$
- 2192 $y = a + bx \ln x + cx\sqrt{x} + dx^2 \ln x$
- 2193 $y = a + bx \ln x + cx\sqrt{x} + dx^2 \sqrt{x}$
- 2194 $y = a + bx \ln x + cx\sqrt{x} + dx^3$
- 2195 $y = a + bx \ln x + cx\sqrt{x} + d e^x$
- 2196 $y = a + bx \ln x + cx\sqrt{x} + d\sqrt{x} \ln x$
- 2197 $y = a + bx \ln x + cx\sqrt{x} + d(\ln x)^2$
- 2198 $y = a + bx \ln x + cx\sqrt{x} + d\frac{x}{\ln x}$
- 2199 $y = a + bx \ln x + cx\sqrt{x} + d\sqrt{x}$
- 2200 $y = a + bx \ln x + cx\sqrt{x} + d \ln x$
- 2201 $y = a + bx \ln x + cx\sqrt{x} + \frac{d}{\ln x}$
- 2202 $y = a + bx \ln x + cx\sqrt{x} + \frac{d}{\sqrt{x}}$
- 2203 $y = a + bx \ln x + cx\sqrt{x} + d\frac{\ln x}{x}$
- 2204 $y = a + bx \ln x + cx\sqrt{x} + \frac{d}{x}$
- 2205 $y = a + bx \ln x + cx\sqrt{x} + \frac{d}{x\sqrt{x}}$
- 2206 $y = a + bx \ln x + cx\sqrt{x} + d\frac{\ln x}{x^2}$
- 2207 $y = a + bx \ln x + cx\sqrt{x} + \frac{d}{x^2}$
- 2208 $y = a + bx \ln x + cx\sqrt{x} + d e^{-x}$
- 2209 $y = a + bx \ln x + cx^2 + dx^2 \ln x$
- 2210 $y = a + bx \ln x + cx^2 + dx^2 \sqrt{x}$
- 2211 $y = a + bx \ln x + cx^2 + dx^3$
- 2212 $y = a + bx \ln x + cx^2 + d e^x$
- 2213 $y = a + bx \ln x + cx^2 + d\sqrt{x} \ln x$
- 2214 $y = a + bx \ln x + cx^2 + d(\ln x)^2$
- 2215 $y = a + bx \ln x + cx^2 + d\frac{x}{\ln x}$
- 2216 $y = a + bx \ln x + cx^2 + d\sqrt{x}$
- 2217 $y = a + bx \ln x + cx^2 + d \ln x$
- 2218 $y = a + bx \ln x + cx^2 + \frac{d}{\ln x}$
- 2219 $y = a + bx \ln x + cx^2 + \frac{d}{\sqrt{x}}$
- 2220 $y = a + bx \ln x + cx^2 + d\frac{\ln x}{x}$
- 2221 $y = a + bx \ln x + cx^2 + \frac{d}{x}$
- 2222 $y = a + bx \ln x + cx^2 + \frac{d}{x\sqrt{x}}$
- 2223 $y = a + bx \ln x + cx^2 + d\frac{\ln x}{x^2}$
- 2224 $y = a + bx \ln x + cx^2 + \frac{d}{x^2}$
- 2225 $y = a + bx \ln x + cx^2 + d e^{-x}$
- 2226 $y = a + bx \ln x + cx^2 \ln x + dx^2 \sqrt{x}$
- 2227 $y = a + bx \ln x + cx^2 \ln x + dx^3$
- 2228 $y = a + bx \ln x + cx^2 \ln x + d e^x$

2229 $y = a + bx \ln x + cx^2 \ln x + d\sqrt{x} \ln x$	2255 $y = a + bx \ln x + cx^2 \sqrt{x} + \frac{d}{x^2}$	2281 $y = a + bx \ln x + c e^x + d \frac{\ln x}{x^2}$
2230 $y = a + bx \ln x + cx^2 \ln x + d(\ln x)^2$	2256 $y = a + bx \ln x + cx^2 \sqrt{x} + d e^{-x}$	2282 $y = a + bx \ln x + c e^x + \frac{d}{x^2}$
2231 $y = a + bx \ln x + cx^2 \ln x + d \frac{x}{\ln x}$	2257 $y = a + bx \ln x + cx^3 + d e^x$	2283 $y = a + bx \ln x + c e^x + d e^{-x}$
2232 $y = a + bx \ln x + cx^2 \ln x + d\sqrt{x}$	2258 $y = a + bx \ln x + cx^3 + d\sqrt{x} \ln x$	2284 $y = a + bx \ln x + c\sqrt{x} \ln x + d(\ln x)^2$
2233 $y = a + bx \ln x + cx^2 \ln x + d \ln x$	2259 $y = a + bx \ln x + cx^3 + d(\ln x)^2$	2285 $y = a + bx \ln x + c\sqrt{x} \ln x + d \frac{x}{\ln x}$
2234 $y = a + bx \ln x + cx^2 \ln x + \frac{d}{\ln x}$	2260 $y = a + bx \ln x + cx^3 + d \frac{x}{\ln x}$	2286 $y = a + bx \ln x + c\sqrt{x} \ln x + d\sqrt{x}$
2235 $y = a + bx \ln x + cx^2 \ln x + \frac{d}{\sqrt{x}}$	2261 $y = a + bx \ln x + cx^3 + d\sqrt{x}$	2287 $y = a + bx \ln x + c\sqrt{x} \ln x + d \ln x$
2236 $y = a + bx \ln x + cx^2 \ln x + d \frac{\ln x}{x}$	2262 $y = a + bx \ln x + cx^3 + d \ln x$	2288 $y = a + bx \ln x + c\sqrt{x} \ln x + \frac{d}{\ln x}$
2237 $y = a + bx \ln x + cx^2 \ln x + \frac{d}{x}$	2263 $y = a + bx \ln x + cx^3 + \frac{d}{\ln x}$	2289 $y = a + bx \ln x + c\sqrt{x} \ln x + \frac{d}{\sqrt{x}}$
2238 $y = a + bx \ln x + cx^2 \ln x + \frac{d}{x\sqrt{x}}$	2264 $y = a + bx \ln x + cx^3 + \frac{d}{\sqrt{x}}$	2290 $y = a + bx \ln x + c\sqrt{x} \ln x + d \frac{\ln x}{x}$
2239 $y = a + bx \ln x + cx^2 \ln x + d \frac{\ln x}{x^2}$	2265 $y = a + bx \ln x + cx^3 + d \frac{\ln x}{x}$	2291 $y = a + bx \ln x + c\sqrt{x} \ln x + \frac{d}{x}$
2240 $y = a + bx \ln x + cx^2 \ln x + \frac{d}{x^2}$	2266 $y = a + bx \ln x + cx^3 + \frac{d}{x}$	2292 $y = a + bx \ln x + c\sqrt{x} \ln x + \frac{d}{x\sqrt{x}}$
2241 $y = a + bx \ln x + cx^2 \ln x + d e^{-x}$	2267 $y = a + bx \ln x + cx^3 + \frac{d}{x\sqrt{x}}$	2293 $y = a + bx \ln x + c\sqrt{x} \ln x + d \frac{\ln x}{x^2}$
2242 $y = a + bx \ln x + cx^2 \sqrt{x} + dx^3$	2268 $y = a + bx \ln x + cx^3 + d \frac{\ln x}{x^2}$	2294 $y = a + bx \ln x + c\sqrt{x} \ln x + \frac{d}{x^2}$
2243 $y = a + bx \ln x + cx^2 \sqrt{x} + d e^x$	2269 $y = a + bx \ln x + cx^3 + \frac{d}{x^2}$	2295 $y = a + bx \ln x + c\sqrt{x} \ln x + d e^{-x}$
2244 $y = a + bx \ln x + cx^2 \sqrt{x} + d\sqrt{x} \ln x$	2270 $y = a + bx \ln x + cx^3 + d e^{-x}$	2296 $y = a + bx \ln x + c(\ln x)^2 + d \frac{x}{\ln x}$
2245 $y = a + bx \ln x + cx^2 \sqrt{x} + d(\ln x)^2$	2271 $y = a + bx \ln x + c e^x + d\sqrt{x} \ln x$	2297 $y = a + bx \ln x + c(\ln x)^2 + d\sqrt{x}$
2246 $y = a + bx \ln x + cx^2 \sqrt{x} + d \frac{x}{\ln x}$	2272 $y = a + bx \ln x + c e^x + d(\ln x)^2$	2298 $y = a + bx \ln x + c(\ln x)^2 + d \ln x$
2247 $y = a + bx \ln x + cx^2 \sqrt{x} + d\sqrt{x}$	2273 $y = a + bx \ln x + c e^x + d \frac{x}{\ln x}$	2299 $y = a + bx \ln x + c(\ln x)^2 + \frac{d}{\ln x}$
2248 $y = a + bx \ln x + cx^2 \sqrt{x} + d \ln x$	2274 $y = a + bx \ln x + c e^x + d\sqrt{x}$	2300 $y = a + bx \ln x + c(\ln x)^2 + \frac{d}{\sqrt{x}}$
2249 $y = a + bx \ln x + cx^2 \sqrt{x} + \frac{d}{\ln x}$	2275 $y = a + bx \ln x + c e^x + d \ln x$	2301 $y = a + bx \ln x + c(\ln x)^2 + d \frac{\ln x}{x}$
2250 $y = a + bx \ln x + cx^2 \sqrt{x} + \frac{d}{\sqrt{x}}$	2276 $y = a + bx \ln x + c e^x + \frac{d}{\ln x}$	2302 $y = a + bx \ln x + c(\ln x)^2 + \frac{d}{x}$
2251 $y = a + bx \ln x + cx^2 \sqrt{x} + d \frac{\ln x}{x}$	2277 $y = a + bx \ln x + c e^x + \frac{d}{\sqrt{x}}$	2303 $y = a + bx \ln x + c(\ln x)^2 + \frac{d}{x\sqrt{x}}$
2252 $y = a + bx \ln x + cx^2 \sqrt{x} + \frac{d}{x}$	2278 $y = a + bx \ln x + c e^x + d \frac{\ln x}{x}$	2304 $y = a + bx \ln x + c(\ln x)^2 + d \frac{\ln x}{x^2}$
2253 $y = a + bx \ln x + cx^2 \sqrt{x} + \frac{d}{x\sqrt{x}}$	2279 $y = a + bx \ln x + c e^x + \frac{d}{x}$	2305 $y = a + bx \ln x + c(\ln x)^2 + \frac{d}{x^2}$
2254 $y = a + bx \ln x + cx^2 \sqrt{x} + d \frac{\ln x}{x^2}$	2280 $y = a + bx \ln x + c e^x + \frac{d}{x\sqrt{x}}$	2306 $y = a + bx \ln x + c(\ln x)^2 + d e^{-x}$

$$\begin{aligned} 2307 \quad y &= a + bx \ln x + c \frac{x}{\ln x} + d\sqrt{x} \\ 2308 \quad y &= a + bx \ln x + c \frac{x}{\ln x} + d \ln x \\ 2309 \quad y &= a + bx \ln x + c \frac{x}{\ln x} + \frac{d}{\ln x} \\ 2310 \quad y &= a + bx \ln x + c \frac{x}{\ln x} + \frac{d}{\sqrt{x}} \\ 2311 \quad y &= a + bx \ln x + c \frac{x}{\ln x} + d \frac{\ln x}{x} \\ 2312 \quad y &= a + bx \ln x + c \frac{x}{\ln x} + \frac{d}{x} \\ 2313 \quad y &= a + bx \ln x + c \frac{x}{\ln x} + \frac{d}{x\sqrt{x}} \\ 2314 \quad y &= a + bx \ln x + c \frac{x}{\ln x} + d \frac{\ln x}{x^2} \\ 2315 \quad y &= a + bx \ln x + c \frac{x}{\ln x} + \frac{d}{x^2} \\ 2316 \quad y &= a + bx \ln x + c \frac{x}{\ln x} + d e^{-x} \\ 2317 \quad y &= a + bx \ln x + c\sqrt{x} + d \ln x \\ 2318 \quad y &= a + bx \ln x + c\sqrt{x} + \frac{d}{\ln x} \\ 2319 \quad y &= a + bx \ln x + c\sqrt{x} + \frac{d}{\sqrt{x}} \\ 2320 \quad y &= a + bx \ln x + c\sqrt{x} + d \frac{\ln x}{x} \\ 2321 \quad y &= a + bx \ln x + c\sqrt{x} + \frac{d}{x} \\ 2322 \quad y &= a + bx \ln x + c\sqrt{x} + \frac{d}{x\sqrt{x}} \\ 2323 \quad y &= a + bx \ln x + c\sqrt{x} + d \frac{\ln x}{x^2} \\ 2324 \quad y &= a + bx \ln x + c\sqrt{x} + \frac{d}{x^2} \\ 2325 \quad y &= a + bx \ln x + c\sqrt{x} + d e^{-x} \\ 2326 \quad y &= a + bx \ln x + c \ln x + \frac{d}{\ln x} \\ 2327 \quad y &= a + bx \ln x + c \ln x + \frac{d}{\sqrt{x}} \\ 2328 \quad y &= a + bx \ln x + c \ln x + d \frac{\ln x}{x} \\ 2329 \quad y &= a + bx \ln x + c \ln x + \frac{d}{x} \\ 2330 \quad y &= a + bx \ln x + c \ln x + \frac{d}{x\sqrt{x}} \\ 2331 \quad y &= a + bx \ln x + c \ln x + d \frac{\ln x}{x^2} \\ 2332 \quad y &= a + bx \ln x + c \ln x + \frac{d}{x^2} \end{aligned}$$

$$\begin{aligned} 2333 \quad y &= a + bx \ln x + c \ln x + d e^{-x} \\ 2334 \quad y &= a + bx \ln x + \frac{c}{\ln x} + \frac{d}{\sqrt{x}} \\ 2335 \quad y &= a + bx \ln x + \frac{c}{\ln x} + d \frac{\ln x}{x} \\ 2336 \quad y &= a + bx \ln x + \frac{c}{\ln x} + \frac{d}{x} \\ 2337 \quad y &= a + bx \ln x + \frac{c}{\ln x} + \frac{d}{x\sqrt{x}} \\ 2338 \quad y &= a + bx \ln x + \frac{c}{\ln x} + d \frac{\ln x}{x^2} \\ 2339 \quad y &= a + bx \ln x + \frac{c}{\ln x} + \frac{d}{x^2} \\ 2340 \quad y &= a + bx \ln x + \frac{c}{\ln x} + d e^{-x} \\ 2341 \quad y &= a + bx \ln x + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x} \\ 2342 \quad y &= a + bx \ln x + \frac{c}{\sqrt{x}} + \frac{d}{x} \\ 2343 \quad y &= a + bx \ln x + \frac{c}{\sqrt{x}} + \frac{d}{x\sqrt{x}} \\ 2344 \quad y &= a + bx \ln x + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x^2} \\ 2345 \quad y &= a + bx \ln x + \frac{c}{\sqrt{x}} + \frac{d}{x^2} \\ 2346 \quad y &= a + bx \ln x + \frac{c}{\sqrt{x}} + d e^{-x} \\ 2347 \quad y &= a + bx \ln x + c \frac{\ln x}{x} + \frac{d}{x} \\ 2348 \quad y &= a + bx \ln x + c \frac{\ln x}{x} + \frac{d}{x\sqrt{x}} \\ 2349 \quad y &= a + bx \ln x + c \frac{\ln x}{x} + d \frac{\ln x}{x^2} \\ 2350 \quad y &= a + bx \ln x + c \frac{\ln x}{x} + \frac{d}{x^2} \\ 2351 \quad y &= a + bx \ln x + c \frac{\ln x}{x} + d e^{-x} \\ 2352 \quad y &= a + bx \ln x + \frac{c}{x} + \frac{d}{x\sqrt{x}} \\ 2353 \quad y &= a + bx \ln x + \frac{c}{x} + d \frac{\ln x}{x^2} \\ 2354 \quad y &= a + bx \ln x + \frac{c}{x} + \frac{d}{x^2} \\ 2355 \quad y &= a + bx \ln x + \frac{c}{x} + d e^{-x} \\ 2356 \quad y &= a + bx \ln x + \frac{c}{x\sqrt{x}} + d \frac{\ln x}{x^2} \\ 2357 \quad y &= a + bx \ln x + \frac{c}{x\sqrt{x}} + \frac{d}{x^2} \\ 2358 \quad y &= a + bx \ln x + \frac{c}{x\sqrt{x}} + d e^{-x} \end{aligned}$$

$$\begin{aligned} 2359 \quad y &= a + bx \ln x + c \frac{\ln x}{x^2} + \frac{d}{x^2} \\ 2360 \quad y &= a + bx \ln x + c \frac{\ln x}{x^2} + d e^{-x} \\ 2361 \quad y &= a + bx \ln x + \frac{c}{x^2} + d e^{-x} \\ 2362 \quad y &= a + bx\sqrt{x} + cx^2 + dx^2 \ln x \\ 2363 \quad y &= a + bx\sqrt{x} + cx^2 + dx^2 \sqrt{x} \\ 2364 \quad y &= a + bx\sqrt{x} + cx^2 + dx^3 \\ 2365 \quad y &= a + bx\sqrt{x} + cx^2 + d e^x \\ 2366 \quad y &= a + bx\sqrt{x} + cx^2 + d\sqrt{x} \ln x \\ 2367 \quad y &= a + bx\sqrt{x} + cx^2 + d(\ln x)^2 \\ 2368 \quad y &= a + bx\sqrt{x} + cx^2 + d \frac{x}{\ln x} \\ 2369 \quad y &= a + bx\sqrt{x} + cx^2 + d\sqrt{x} \\ 2370 \quad y &= a + bx\sqrt{x} + cx^2 + d \ln x \\ 2371 \quad y &= a + bx\sqrt{x} + cx^2 + \frac{d}{\ln x} \\ 2372 \quad y &= a + bx\sqrt{x} + cx^2 + \frac{d}{\sqrt{x}} \\ 2373 \quad y &= a + bx\sqrt{x} + cx^2 + d \frac{\ln x}{x} \\ 2374 \quad y &= a + bx\sqrt{x} + cx^2 + \frac{d}{x} \\ 2375 \quad y &= a + bx\sqrt{x} + cx^2 + \frac{d}{x\sqrt{x}} \\ 2376 \quad y &= a + bx\sqrt{x} + cx^2 + d \frac{\ln x}{x^2} \\ 2377 \quad y &= a + bx\sqrt{x} + cx^2 + \frac{d}{x^2} \\ 2378 \quad y &= a + bx\sqrt{x} + cx^2 + d e^{-x} \\ 2379 \quad y &= a + bx\sqrt{x} + cx^2 \ln x + dx^2 \sqrt{x} \\ 2380 \quad y &= a + bx\sqrt{x} + cx^2 \ln x + dx^3 \\ 2381 \quad y &= a + bx\sqrt{x} + cx^2 \ln x + d e^x \\ 2382 \quad y &= a + bx\sqrt{x} + cx^2 \ln x + d\sqrt{x} \ln x \\ 2383 \quad y &= a + bx\sqrt{x} + cx^2 \ln x + d(\ln x)^2 \\ 2384 \quad y &= a + bx\sqrt{x} + cx^2 \ln x + d \frac{x}{\ln x} \end{aligned}$$

2385 $y = a + bx\sqrt{x} + cx^2 \ln x + d\sqrt{x}$	2411 $y = a + bx\sqrt{x} + cx^3 + d\sqrt{x} \ln x$	2437 $y = a + bx\sqrt{x} + c\sqrt{x} \ln x + d(\ln x)^2$
2386 $y = a + bx\sqrt{x} + cx^2 \ln x + d \ln x$	2412 $y = a + bx\sqrt{x} + cx^3 + d(\ln x)^2$	2438 $y = a + bx\sqrt{x} + c\sqrt{x} \ln x + d \frac{x}{\ln x}$
2387 $y = a + bx\sqrt{x} + cx^2 \ln x + \frac{d}{\ln x}$	2413 $y = a + bx\sqrt{x} + cx^3 + d \frac{x}{\ln x}$	2439 $y = a + bx\sqrt{x} + c\sqrt{x} \ln x + d\sqrt{x}$
2388 $y = a + bx\sqrt{x} + cx^2 \ln x + \frac{d}{\sqrt{x}}$	2414 $y = a + bx\sqrt{x} + cx^3 + d\sqrt{x}$	2440 $y = a + bx\sqrt{x} + c\sqrt{x} \ln x + d \ln x$
2389 $y = a + bx\sqrt{x} + cx^2 \ln x + d \frac{\ln x}{x}$	2415 $y = a + bx\sqrt{x} + cx^3 + d \ln x$	2441 $y = a + bx\sqrt{x} + c\sqrt{x} \ln x + \frac{d}{\ln x}$
2390 $y = a + bx\sqrt{x} + cx^2 \ln x + \frac{d}{x}$	2416 $y = a + bx\sqrt{x} + cx^3 + \frac{d}{\ln x}$	2442 $y = a + bx\sqrt{x} + c\sqrt{x} \ln x + \frac{d}{\sqrt{x}}$
2391 $y = a + bx\sqrt{x} + cx^2 \ln x + \frac{d}{x\sqrt{x}}$	2417 $y = a + bx\sqrt{x} + cx^3 + \frac{d}{\sqrt{x}}$	2443 $y = a + bx\sqrt{x} + c\sqrt{x} \ln x + d \frac{\ln x}{x}$
2392 $y = a + bx\sqrt{x} + cx^2 \ln x + d \frac{\ln x}{x^2}$	2418 $y = a + bx\sqrt{x} + cx^3 + d \frac{\ln x}{x}$	2444 $y = a + bx\sqrt{x} + c\sqrt{x} \ln x + \frac{d}{x}$
2393 $y = a + bx\sqrt{x} + cx^2 \ln x + \frac{d}{x^2}$	2419 $y = a + bx\sqrt{x} + cx^3 + \frac{d}{x}$	2445 $y = a + bx\sqrt{x} + c\sqrt{x} \ln x + \frac{d}{x\sqrt{x}}$
2394 $y = a + bx\sqrt{x} + cx^2 \ln x + d e^{-x}$	2420 $y = a + bx\sqrt{x} + cx^3 + \frac{d}{x\sqrt{x}}$	2446 $y = a + bx\sqrt{x} + c\sqrt{x} \ln x + d \frac{\ln x}{x^2}$
2395 $y = a + bx\sqrt{x} + cx^2 \sqrt{x} + dx^3$	2421 $y = a + bx\sqrt{x} + cx^3 + d \frac{\ln x}{x^2}$	2447 $y = a + bx\sqrt{x} + c\sqrt{x} \ln x + \frac{d}{x^2}$
2396 $y = a + bx\sqrt{x} + cx^2 \sqrt{x} + d e^x$	2422 $y = a + bx\sqrt{x} + cx^3 + \frac{d}{x^2}$	2448 $y = a + bx\sqrt{x} + c\sqrt{x} \ln x + d e^{-x}$
2397 $y = a + bx\sqrt{x} + cx^2 \sqrt{x} + d\sqrt{x} \ln x$	2423 $y = a + bx\sqrt{x} + cx^3 + d e^{-x}$	2449 $y = a + bx\sqrt{x} + c(\ln x)^2 + d \frac{x}{\ln x}$
2398 $y = a + bx\sqrt{x} + cx^2 \sqrt{x} + d(\ln x)^2$	2424 $y = a + bx\sqrt{x} + c e^x + d\sqrt{x} \ln x$	2450 $y = a + bx\sqrt{x} + c(\ln x)^2 + d\sqrt{x}$
2399 $y = a + bx\sqrt{x} + cx^2 \sqrt{x} + d \frac{x}{\ln x}$	2425 $y = a + bx\sqrt{x} + c e^x + d(\ln x)^2$	2451 $y = a + bx\sqrt{x} + c(\ln x)^2 + d \ln x$
2400 $y = a + bx\sqrt{x} + cx^2 \sqrt{x} + d\sqrt{x}$	2426 $y = a + bx\sqrt{x} + c e^x + d \frac{x}{\ln x}$	2452 $y = a + bx\sqrt{x} + c(\ln x)^2 + \frac{d}{\ln x}$
2401 $y = a + bx\sqrt{x} + cx^2 \sqrt{x} + d \ln x$	2427 $y = a + bx\sqrt{x} + c e^x + d\sqrt{x}$	2453 $y = a + bx\sqrt{x} + c(\ln x)^2 + \frac{d}{\sqrt{x}}$
2402 $y = a + bx\sqrt{x} + cx^2 \sqrt{x} + \frac{d}{\ln x}$	2428 $y = a + bx\sqrt{x} + c e^x + d \ln x$	2454 $y = a + bx\sqrt{x} + c(\ln x)^2 + d \frac{\ln x}{x}$
2403 $y = a + bx\sqrt{x} + cx^2 \sqrt{x} + \frac{d}{\sqrt{x}}$	2429 $y = a + bx\sqrt{x} + c e^x + \frac{d}{\ln x}$	2455 $y = a + bx\sqrt{x} + c(\ln x)^2 + \frac{d}{x}$
2404 $y = a + bx\sqrt{x} + cx^2 \sqrt{x} + d \frac{\ln x}{x}$	2430 $y = a + bx\sqrt{x} + c e^x + \frac{d}{\sqrt{x}}$	2456 $y = a + bx\sqrt{x} + c(\ln x)^2 + \frac{d}{x\sqrt{x}}$
2405 $y = a + bx\sqrt{x} + cx^2 \sqrt{x} + \frac{d}{x}$	2431 $y = a + bx\sqrt{x} + c e^x + d \frac{\ln x}{x}$	2457 $y = a + bx\sqrt{x} + c(\ln x)^2 + d \frac{\ln x}{x^2}$
2406 $y = a + bx\sqrt{x} + cx^2 \sqrt{x} + \frac{d}{x\sqrt{x}}$	2432 $y = a + bx\sqrt{x} + c e^x + \frac{d}{x}$	2458 $y = a + bx\sqrt{x} + c(\ln x)^2 + \frac{d}{x^2}$
2407 $y = a + bx\sqrt{x} + cx^2 \sqrt{x} + d \frac{\ln x}{x^2}$	2433 $y = a + bx\sqrt{x} + c e^x + \frac{d}{x\sqrt{x}}$	2459 $y = a + bx\sqrt{x} + c(\ln x)^2 + d e^{-x}$
2408 $y = a + bx\sqrt{x} + cx^2 \sqrt{x} + \frac{d}{x^2}$	2434 $y = a + bx\sqrt{x} + c e^x + d \frac{\ln x}{x^2}$	2460 $y = a + bx\sqrt{x} + c \frac{x}{\ln x} + d\sqrt{x}$
2409 $y = a + bx\sqrt{x} + cx^2 \sqrt{x} + d e^{-x}$	2435 $y = a + bx\sqrt{x} + c e^x + \frac{d}{x^2}$	2461 $y = a + bx\sqrt{x} + c \frac{x}{\ln x} + d \ln x$
2410 $y = a + bx\sqrt{x} + cx^3 + d e^x$	2436 $y = a + bx\sqrt{x} + c e^x + d e^{-x}$	2462 $y = a + bx\sqrt{x} + c \frac{x}{\ln x} + \frac{d}{\ln x}$

- 2463 $y = a + bx\sqrt{x} + c \frac{x}{\ln x} + \frac{d}{\sqrt{x}}$
- 2464 $y = a + bx\sqrt{x} + c \frac{x}{\ln x} + d \frac{\ln x}{x}$
- 2465 $y = a + bx\sqrt{x} + c \frac{x}{\ln x} + \frac{d}{x}$
- 2466 $y = a + bx\sqrt{x} + c \frac{x}{\ln x} + \frac{d}{x\sqrt{x}}$
- 2467 $y = a + bx\sqrt{x} + c \frac{x}{\ln x} + d \frac{\ln x}{x^2}$
- 2468 $y = a + bx\sqrt{x} + c \frac{x}{\ln x} + \frac{d}{x^2}$
- 2469 $y = a + bx\sqrt{x} + c \frac{x}{\ln x} + d e^{-x}$
- 2470 $y = a + bx\sqrt{x} + c\sqrt{x} + d \ln x$
- 2471 $y = a + bx\sqrt{x} + c\sqrt{x} + \frac{d}{\ln x}$
- 2472 $y = a + bx\sqrt{x} + c\sqrt{x} + \frac{d}{\sqrt{x}}$
- 2473 $y = a + bx\sqrt{x} + c\sqrt{x} + d \frac{\ln x}{x}$
- 2474 $y = a + bx\sqrt{x} + c\sqrt{x} + \frac{d}{x}$
- 2475 $y = a + bx\sqrt{x} + c\sqrt{x} + \frac{d}{x\sqrt{x}}$
- 2476 $y = a + bx\sqrt{x} + c\sqrt{x} + d \frac{\ln x}{x^2}$
- 2477 $y = a + bx\sqrt{x} + c\sqrt{x} + \frac{d}{x^2}$
- 2478 $y = a + bx\sqrt{x} + c\sqrt{x} + d e^{-x}$
- 2479 $y = a + bx\sqrt{x} + c \ln x + \frac{d}{\ln x}$
- 2480 $y = a + bx\sqrt{x} + c \ln x + \frac{d}{\sqrt{x}}$
- 2481 $y = a + bx\sqrt{x} + c \ln x + d \frac{\ln x}{x}$
- 2482 $y = a + bx\sqrt{x} + c \ln x + \frac{d}{x}$
- 2483 $y = a + bx\sqrt{x} + c \ln x + \frac{d}{x\sqrt{x}}$
- 2484 $y = a + bx\sqrt{x} + c \ln x + d \frac{\ln x}{x^2}$
- 2485 $y = a + bx\sqrt{x} + c \ln x + \frac{d}{x^2}$
- 2486 $y = a + bx\sqrt{x} + c \ln x + d e^{-x}$
- 2487 $y = a + bx\sqrt{x} + \frac{c}{\ln x} + \frac{d}{\sqrt{x}}$
- 2488 $y = a + bx\sqrt{x} + \frac{c}{\ln x} + d \frac{\ln x}{x}$
- 2489 $y = a + bx\sqrt{x} + \frac{c}{\ln x} + \frac{d}{x}$
- 2490 $y = a + bx\sqrt{x} + \frac{c}{\ln x} + \frac{d}{x\sqrt{x}}$
- 2491 $y = a + bx\sqrt{x} + \frac{c}{\ln x} + d \frac{\ln x}{x^2}$
- 2492 $y = a + bx\sqrt{x} + \frac{c}{\ln x} + \frac{d}{x^2}$
- 2493 $y = a + bx\sqrt{x} + \frac{c}{\ln x} + d e^{-x}$
- 2494 $y = a + bx\sqrt{x} + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x}$
- 2495 $y = a + bx\sqrt{x} + \frac{c}{\sqrt{x}} + \frac{d}{x}$
- 2496 $y = a + bx\sqrt{x} + \frac{c}{\sqrt{x}} + \frac{d}{x\sqrt{x}}$
- 2497 $y = a + bx\sqrt{x} + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x^2}$
- 2498 $y = a + bx\sqrt{x} + \frac{c}{\sqrt{x}} + \frac{d}{x^2}$
- 2499 $y = a + bx\sqrt{x} + \frac{c}{\sqrt{x}} + d e^{-x}$
- 2500 $y = a + bx\sqrt{x} + c \frac{\ln x}{x} + \frac{d}{x}$
- 2501 $y = a + bx\sqrt{x} + c \frac{\ln x}{x} + \frac{d}{x\sqrt{x}}$
- 2502 $y = a + bx\sqrt{x} + c \frac{\ln x}{x} + d \frac{\ln x}{x^2}$
- 2503 $y = a + bx\sqrt{x} + c \frac{\ln x}{x} + \frac{d}{x^2}$
- 2504 $y = a + bx\sqrt{x} + c \frac{\ln x}{x} + d e^{-x}$
- 2505 $y = a + bx\sqrt{x} + \frac{c}{x} + \frac{d}{x\sqrt{x}}$
- 2506 $y = a + bx\sqrt{x} + \frac{c}{x} + d \frac{\ln x}{x^2}$
- 2507 $y = a + bx\sqrt{x} + \frac{c}{x} + \frac{d}{x^2}$
- 2508 $y = a + bx\sqrt{x} + \frac{c}{x} + d e^{-x}$
- 2509 $y = a + bx\sqrt{x} + \frac{c}{x\sqrt{x}} + d \frac{\ln x}{x^2}$
- 2510 $y = a + bx\sqrt{x} + \frac{c}{x\sqrt{x}} + \frac{d}{x^2}$
- 2511 $y = a + bx\sqrt{x} + \frac{c}{x\sqrt{x}} + d e^{-x}$
- 2512 $y = a + bx\sqrt{x} + c \frac{\ln x}{x^2} + \frac{d}{x^2}$
- 2513 $y = a + bx\sqrt{x} + c \frac{\ln x}{x^2} + d e^{-x}$
- 2514 $y = a + bx\sqrt{x} + \frac{c}{x^2} + d e^{-x}$
- 2515 $y = a + bx^2 + cx^2 \ln x + dx^2 \sqrt{x}$
- 2516 $y = a + bx^2 + cx^2 \ln x + dx^3$
- 2517 $y = a + bx^2 + cx^2 \ln x + d e^x$
- 2518 $y = a + bx^2 + cx^2 \ln x + d\sqrt{x} \ln x$
- 2519 $y = a + bx^2 + cx^2 \ln x + d(\ln x)^2$
- 2520 $y = a + bx^2 + cx^2 \ln x + d \frac{x}{\ln x}$
- 2521 $y = a + bx^2 + cx^2 \ln x + d\sqrt{x}$
- 2522 $y = a + bx^2 + cx^2 \ln x + d \ln x$
- 2523 $y = a + bx^2 + cx^2 \ln x + \frac{d}{\ln x}$
- 2524 $y = a + bx^2 + cx^2 \ln x + \frac{d}{\sqrt{x}}$
- 2525 $y = a + bx^2 + cx^2 \ln x + d \frac{\ln x}{x}$
- 2526 $y = a + bx^2 + cx^2 \ln x + \frac{d}{x}$
- 2527 $y = a + bx^2 + cx^2 \ln x + \frac{d}{x\sqrt{x}}$
- 2528 $y = a + bx^2 + cx^2 \ln x + d \frac{\ln x}{x^2}$
- 2529 $y = a + bx^2 + cx^2 \ln x + \frac{d}{x^2}$
- 2530 $y = a + bx^2 + cx^2 \ln x + d e^{-x}$
- 2531 $y = a + bx^2 + cx^2 \sqrt{x} + dx^3$
- 2532 $y = a + bx^2 + cx^2 \sqrt{x} + d e^x$
- 2533 $y = a + bx^2 + cx^2 \sqrt{x} + d\sqrt{x} \ln x$
- 2534 $y = a + bx^2 + cx^2 \sqrt{x} + d(\ln x)^2$
- 2535 $y = a + bx^2 + cx^2 \sqrt{x} + d \frac{x}{\ln x}$
- 2536 $y = a + bx^2 + cx^2 \sqrt{x} + d\sqrt{x}$
- 2537 $y = a + bx^2 + cx^2 \sqrt{x} + d \ln x$
- 2538 $y = a + bx^2 + cx^2 \sqrt{x} + \frac{d}{\ln x}$
- 2539 $y = a + bx^2 + cx^2 \sqrt{x} + \frac{d}{\sqrt{x}}$
- 2540 $y = a + bx^2 + cx^2 \sqrt{x} + d \frac{\ln x}{x}$

- 2541 $y = a + bx^2 + cx^2\sqrt{x} + \frac{d}{x}$
- 2542 $y = a + bx^2 + cx^2\sqrt{x} + \frac{d}{x\sqrt{x}}$
- 2543 $y = a + bx^2 + cx^2\sqrt{x} + d\frac{\ln x}{x^2}$
- 2544 $y = a + bx^2 + cx^2\sqrt{x} + \frac{d}{x^2}$
- 2545 $y = a + bx^2 + cx^2\sqrt{x} + de^{-x}$
- 2546 $y = a + bx^2 + cx^3 + de^x$
- 2547 $y = a + bx^2 + cx^3 + d\sqrt{x}\ln x$
- 2548 $y = a + bx^2 + cx^3 + d(\ln x)^2$
- 2549 $y = a + bx^2 + cx^3 + d\frac{x}{\ln x}$
- 2550 $y = a + bx^2 + cx^3 + d\sqrt{x}$
- 2551 $y = a + bx^2 + cx^3 + d\ln x$
- 2552 $y = a + bx^2 + cx^3 + \frac{d}{\ln x}$
- 2553 $y = a + bx^2 + cx^3 + \frac{d}{\sqrt{x}}$
- 2554 $y = a + bx^2 + cx^3 + d\frac{\ln x}{x}$
- 2555 $y = a + bx^2 + cx^3 + \frac{d}{x}$
- 2556 $y = a + bx^2 + cx^3 + \frac{d}{x\sqrt{x}}$
- 2557 $y = a + bx^2 + cx^3 + d\frac{\ln x}{x^2}$
- 2558 $y = a + bx^2 + cx^3 + \frac{d}{x^2}$
- 2559 $y = a + bx^2 + cx^3 + de^{-x}$
- 2560 $y = a + bx^2 + ce^x + d\sqrt{x}\ln x$
- 2561 $y = a + bx^2 + ce^x + d(\ln x)^2$
- 2562 $y = a + bx^2 + ce^x + d\frac{x}{\ln x}$
- 2563 $y = a + bx^2 + ce^x + d\sqrt{x}$
- 2564 $y = a + bx^2 + ce^x + d\ln x$
- 2565 $y = a + bx^2 + ce^x + \frac{d}{\ln x}$
- 2566 $y = a + bx^2 + ce^x + \frac{d}{\sqrt{x}}$
- 2567 $y = a + bx^2 + ce^x + d\frac{\ln x}{x}$
- 2568 $y = a + bx^2 + ce^x + \frac{d}{x}$
- 2569 $y = a + bx^2 + ce^x + \frac{d}{x\sqrt{x}}$
- 2570 $y = a + bx^2 + ce^x + d\frac{\ln x}{x^2}$
- 2571 $y = a + bx^2 + ce^x + \frac{d}{x^2}$
- 2572 $y = a + bx^2 + ce^x + de^{-x}$
- 2573 $y = a + bx^2 + c\sqrt{x}\ln x + d(\ln x)^2$
- 2574 $y = a + bx^2 + c\sqrt{x}\ln x + d\frac{x}{\ln x}$
- 2575 $y = a + bx^2 + c\sqrt{x}\ln x + d\sqrt{x}$
- 2576 $y = a + bx^2 + c\sqrt{x}\ln x + d\ln x$
- 2577 $y = a + bx^2 + c\sqrt{x}\ln x + \frac{d}{\ln x}$
- 2578 $y = a + bx^2 + c\sqrt{x}\ln x + \frac{d}{\sqrt{x}}$
- 2579 $y = a + bx^2 + c\sqrt{x}\ln x + d\frac{\ln x}{x}$
- 2580 $y = a + bx^2 + c\sqrt{x}\ln x + \frac{d}{x}$
- 2581 $y = a + bx^2 + c\sqrt{x}\ln x + \frac{d}{x\sqrt{x}}$
- 2582 $y = a + bx^2 + c\sqrt{x}\ln x + d\frac{\ln x}{x^2}$
- 2583 $y = a + bx^2 + c\sqrt{x}\ln x + \frac{d}{x^2}$
- 2584 $y = a + bx^2 + c\sqrt{x}\ln x + de^{-x}$
- 2585 $y = a + bx^2 + c(\ln x)^2 + d\frac{x}{\ln x}$
- 2586 $y = a + bx^2 + c(\ln x)^2 + d\sqrt{x}$
- 2587 $y = a + bx^2 + c(\ln x)^2 + d\ln x$
- 2588 $y = a + bx^2 + c(\ln x)^2 + \frac{d}{\ln x}$
- 2589 $y = a + bx^2 + c(\ln x)^2 + \frac{d}{\sqrt{x}}$
- 2590 $y = a + bx^2 + c(\ln x)^2 + d\frac{\ln x}{x}$
- 2591 $y = a + bx^2 + c(\ln x)^2 + \frac{d}{x}$
- 2592 $y = a + bx^2 + c(\ln x)^2 + \frac{d}{x\sqrt{x}}$
- 2593 $y = a + bx^2 + c(\ln x)^2 + d\frac{\ln x}{x^2}$
- 2594 $y = a + bx^2 + c(\ln x)^2 + \frac{d}{x^2}$
- 2595 $y = a + bx^2 + c(\ln x)^2 + de^{-x}$
- 2596 $y = a + bx^2 + c\frac{x}{\ln x} + d\sqrt{x}$
- 2597 $y = a + bx^2 + c\frac{x}{\ln x} + d\ln x$
- 2598 $y = a + bx^2 + c\frac{x}{\ln x} + \frac{d}{\ln x}$
- 2599 $y = a + bx^2 + c\frac{x}{\ln x} + \frac{d}{\sqrt{x}}$
- 2600 $y = a + bx^2 + c\frac{x}{\ln x} + d\frac{\ln x}{x}$
- 2601 $y = a + bx^2 + c\frac{x}{\ln x} + \frac{d}{x}$
- 2602 $y = a + bx^2 + c\frac{x}{\ln x} + \frac{d}{x\sqrt{x}}$
- 2603 $y = a + bx^2 + c\frac{x}{\ln x} + d\frac{\ln x}{x^2}$
- 2604 $y = a + bx^2 + c\frac{x}{\ln x} + \frac{d}{x^2}$
- 2605 $y = a + bx^2 + c\frac{x}{\ln x} + de^{-x}$
- 2606 $y = a + bx^2 + c\sqrt{x} + d\ln x$
- 2607 $y = a + bx^2 + c\sqrt{x} + \frac{d}{\ln x}$
- 2608 $y = a + bx^2 + c\sqrt{x} + \frac{d}{\sqrt{x}}$
- 2609 $y = a + bx^2 + c\sqrt{x} + d\frac{\ln x}{x}$
- 2610 $y = a + bx^2 + c\sqrt{x} + \frac{d}{x}$
- 2611 $y = a + bx^2 + c\sqrt{x} + \frac{d}{x\sqrt{x}}$
- 2612 $y = a + bx^2 + c\sqrt{x} + d\frac{\ln x}{x^2}$
- 2613 $y = a + bx^2 + c\sqrt{x} + \frac{d}{x^2}$
- 2614 $y = a + bx^2 + c\sqrt{x} + de^{-x}$
- 2615 $y = a + bx^2 + c\ln x + \frac{d}{\ln x}$
- 2616 $y = a + bx^2 + c\ln x + \frac{d}{\sqrt{x}}$
- 2617 $y = a + bx^2 + c\ln x + d\frac{\ln x}{x}$
- 2618 $y = a + bx^2 + c\ln x + \frac{d}{x}$

$$\begin{aligned}
 2619 \quad y &= a + bx^2 + c \ln x + \frac{d}{x\sqrt{x}} \\
 2620 \quad y &= a + bx^2 + c \ln x + d \frac{\ln x}{x^2} \\
 2621 \quad y &= a + bx^2 + c \ln x + \frac{d}{x^2} \\
 2622 \quad y &= a + bx^2 + c \ln x + d e^{-x} \\
 2623 \quad y &= a + bx^2 + \frac{c}{\ln x} + \frac{d}{\sqrt{x}} \\
 2624 \quad y &= a + bx^2 + \frac{c}{\ln x} + d \frac{\ln x}{x} \\
 2625 \quad y &= a + bx^2 + \frac{c}{\ln x} + \frac{d}{x} \\
 2626 \quad y &= a + bx^2 + \frac{c}{\ln x} + \frac{d}{x\sqrt{x}} \\
 2627 \quad y &= a + bx^2 + \frac{c}{\ln x} + d \frac{\ln x}{x^2} \\
 2628 \quad y &= a + bx^2 + \frac{c}{\ln x} + \frac{d}{x^2} \\
 2629 \quad y &= a + bx^2 + \frac{c}{\ln x} + d e^{-x} \\
 2630 \quad y &= a + bx^2 + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x} \\
 2631 \quad y &= a + bx^2 + \frac{c}{\sqrt{x}} + \frac{d}{x} \\
 2632 \quad y &= a + bx^2 + \frac{c}{\sqrt{x}} + \frac{d}{x\sqrt{x}} \\
 2633 \quad y &= a + bx^2 + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x^2} \\
 2634 \quad y &= a + bx^2 + \frac{c}{\sqrt{x}} + \frac{d}{x^2} \\
 2635 \quad y &= a + bx^2 + \frac{c}{\sqrt{x}} + d e^{-x} \\
 2636 \quad y &= a + bx^2 + c \frac{\ln x}{x} + \frac{d}{x} \\
 2637 \quad y &= a + bx^2 + c \frac{\ln x}{x} + \frac{d}{x\sqrt{x}} \\
 2638 \quad y &= a + bx^2 + c \frac{\ln x}{x} + d \frac{\ln x}{x^2} \\
 2639 \quad y &= a + bx^2 + c \frac{\ln x}{x} + \frac{d}{x^2} \\
 2640 \quad y &= a + bx^2 + c \frac{\ln x}{x} + d e^{-x} \\
 2641 \quad y &= a + bx^2 + \frac{c}{x} + \frac{d}{x\sqrt{x}} \\
 2642 \quad y &= a + bx^2 + \frac{c}{x} + d \frac{\ln x}{x^2} \\
 2643 \quad y &= a + bx^2 + \frac{c}{x} + \frac{d}{x^2} \\
 2644 \quad y &= a + bx^2 + \frac{c}{x} + d e^{-x}
 \end{aligned}$$

$$\begin{aligned}
 2645 \quad y &= a + bx^2 + \frac{c}{x\sqrt{x}} + d \frac{\ln x}{x^2} \\
 2646 \quad y &= a + bx^2 + \frac{c}{x\sqrt{x}} + \frac{d}{x^2} \\
 2647 \quad y &= a + bx^2 + \frac{c}{x\sqrt{x}} + d e^{-x} \\
 2648 \quad y &= a + bx^2 + c \frac{\ln x}{x^2} + \frac{d}{x^2} \\
 2649 \quad y &= a + bx^2 + c \frac{\ln x}{x^2} + d e^{-x} \\
 2650 \quad y &= a + bx^2 + \frac{c}{x^2} + d e^{-x} \\
 2651 \quad y &= a + bx^2 \ln x + cx^2 \sqrt{x} + dx^3 \\
 2652 \quad y &= a + bx^2 \ln x + cx^2 \sqrt{x} + d e^x \\
 2653 \quad y &= a + bx^2 \ln x + cx^2 \sqrt{x} + d \sqrt{x} \ln x \\
 2654 \quad y &= a + bx^2 \ln x + cx^2 \sqrt{x} + d (\ln x)^2 \\
 2655 \quad y &= a + bx^2 \ln x + cx^2 \sqrt{x} + d \frac{x}{\ln x} \\
 2656 \quad y &= a + bx^2 \ln x + cx^2 \sqrt{x} + d \sqrt{x} \\
 2657 \quad y &= a + bx^2 \ln x + cx^2 \sqrt{x} + d \ln x \\
 2658 \quad y &= a + bx^2 \ln x + cx^2 \sqrt{x} + \frac{d}{\ln x} \\
 2659 \quad y &= a + bx^2 \ln x + cx^2 \sqrt{x} + \frac{d}{\sqrt{x}} \\
 2660 \quad y &= a + bx^2 \ln x + cx^2 \sqrt{x} + d \frac{\ln x}{x} \\
 2661 \quad y &= a + bx^2 \ln x + cx^2 \sqrt{x} + \frac{d}{x} \\
 2662 \quad y &= a + bx^2 \ln x + cx^2 \sqrt{x} + \frac{d}{x\sqrt{x}} \\
 2663 \quad y &= a + bx^2 \ln x + cx^2 \sqrt{x} + d \frac{\ln x}{x^2} \\
 2664 \quad y &= a + bx^2 \ln x + cx^2 \sqrt{x} + \frac{d}{x^2} \\
 2665 \quad y &= a + bx^2 \ln x + cx^2 \sqrt{x} + d e^{-x} \\
 2666 \quad y &= a + bx^2 \ln x + cx^3 + d e^x \\
 2667 \quad y &= a + bx^2 \ln x + cx^3 + d \sqrt{x} \ln x \\
 2668 \quad y &= a + bx^2 \ln x + cx^3 + d (\ln x)^2 \\
 2669 \quad y &= a + bx^2 \ln x + cx^3 + d \frac{x}{\ln x} \\
 2670 \quad y &= a + bx^2 \ln x + cx^3 + d \sqrt{x}
 \end{aligned}$$

$$\begin{aligned}
 2671 \quad y &= a + bx^2 \ln x + cx^3 + d \ln x \\
 2672 \quad y &= a + bx^2 \ln x + cx^3 + \frac{d}{\ln x} \\
 2673 \quad y &= a + bx^2 \ln x + cx^3 + \frac{d}{\sqrt{x}} \\
 2674 \quad y &= a + bx^2 \ln x + cx^3 + d \frac{\ln x}{x} \\
 2675 \quad y &= a + bx^2 \ln x + cx^3 + \frac{d}{x} \\
 2676 \quad y &= a + bx^2 \ln x + cx^3 + \frac{d}{x\sqrt{x}} \\
 2677 \quad y &= a + bx^2 \ln x + cx^3 + d \frac{\ln x}{x^2} \\
 2678 \quad y &= a + bx^2 \ln x + cx^3 + \frac{d}{x^2} \\
 2679 \quad y &= a + bx^2 \ln x + cx^3 + d e^{-x} \\
 2680 \quad y &= a + bx^2 \ln x + c e^x + d \sqrt{x} \ln x \\
 2681 \quad y &= a + bx^2 \ln x + c e^x + d (\ln x)^2 \\
 2682 \quad y &= a + bx^2 \ln x + c e^x + d \frac{x}{\ln x} \\
 2683 \quad y &= a + bx^2 \ln x + c e^x + d \sqrt{x} \\
 2684 \quad y &= a + bx^2 \ln x + c e^x + d \ln x \\
 2685 \quad y &= a + bx^2 \ln x + c e^x + \frac{d}{\ln x} \\
 2686 \quad y &= a + bx^2 \ln x + c e^x + \frac{d}{\sqrt{x}} \\
 2687 \quad y &= a + bx^2 \ln x + c e^x + d \frac{\ln x}{x} \\
 2688 \quad y &= a + bx^2 \ln x + c e^x + \frac{d}{x} \\
 2689 \quad y &= a + bx^2 \ln x + c e^x + \frac{d}{x\sqrt{x}} \\
 2690 \quad y &= a + bx^2 \ln x + c e^x + d \frac{\ln x}{x^2} \\
 2691 \quad y &= a + bx^2 \ln x + c e^x + \frac{d}{x^2} \\
 2692 \quad y &= a + bx^2 \ln x + c e^x + d e^{-x} \\
 2693 \quad y &= a + bx^2 \ln x + c \sqrt{x} \ln x + d (\ln x)^2 \\
 2694 \quad y &= a + bx^2 \ln x + c \sqrt{x} \ln x + d \frac{x}{\ln x} \\
 2695 \quad y &= a + bx^2 \ln x + c \sqrt{x} \ln x + d \sqrt{x} \\
 2696 \quad y &= a + bx^2 \ln x + c \sqrt{x} \ln x + d \ln x
 \end{aligned}$$

- 2697 $y = a + bx^2 \ln x + c\sqrt{x} \ln x + \frac{d}{\ln x}$
- 2698 $y = a + bx^2 \ln x + c\sqrt{x} \ln x + \frac{d}{\sqrt{x}}$
- 2699 $y = a + bx^2 \ln x + c\sqrt{x} \ln x + d \frac{\ln x}{x}$
- 2700 $y = a + bx^2 \ln x + c\sqrt{x} \ln x + \frac{d}{x}$
- 2701 $y = a + bx^2 \ln x + c\sqrt{x} \ln x + \frac{d}{x\sqrt{x}}$
- 2702 $y = a + bx^2 \ln x + c\sqrt{x} \ln x + d \frac{\ln x}{x^2}$
- 2703 $y = a + bx^2 \ln x + c\sqrt{x} \ln x + \frac{d}{x^2}$
- 2704 $y = a + bx^2 \ln x + c\sqrt{x} \ln x + d e^{-x}$
- 2705 $y = a + bx^2 \ln x + c(\ln x)^2 + d \frac{x}{\ln x}$
- 2706 $y = a + bx^2 \ln x + c(\ln x)^2 + d\sqrt{x}$
- 2707 $y = a + bx^2 \ln x + c(\ln x)^2 + d \ln x$
- 2708 $y = a + bx^2 \ln x + c(\ln x)^2 + \frac{d}{\ln x}$
- 2709 $y = a + bx^2 \ln x + c(\ln x)^2 + \frac{d}{\sqrt{x}}$
- 2710 $y = a + bx^2 \ln x + c(\ln x)^2 + d \frac{\ln x}{x}$
- 2711 $y = a + bx^2 \ln x + c(\ln x)^2 + \frac{d}{x}$
- 2712 $y = a + bx^2 \ln x + c(\ln x)^2 + \frac{d}{x\sqrt{x}}$
- 2713 $y = a + bx^2 \ln x + c(\ln x)^2 + d \frac{\ln x}{x^2}$
- 2714 $y = a + bx^2 \ln x + c(\ln x)^2 + \frac{d}{x^2}$
- 2715 $y = a + bx^2 \ln x + c(\ln x)^2 + d e^{-x}$
- 2716 $y = a + bx^2 \ln x + c \frac{x}{\ln x} + d\sqrt{x}$
- 2717 $y = a + bx^2 \ln x + c \frac{x}{\ln x} + d \ln x$
- 2718 $y = a + bx^2 \ln x + c \frac{x}{\ln x} + \frac{d}{\ln x}$
- 2719 $y = a + bx^2 \ln x + c \frac{x}{\ln x} + \frac{d}{\sqrt{x}}$
- 2720 $y = a + bx^2 \ln x + c \frac{x}{\ln x} + d \frac{\ln x}{x}$
- 2721 $y = a + bx^2 \ln x + c \frac{x}{\ln x} + \frac{d}{x}$
- 2722 $y = a + bx^2 \ln x + c \frac{x}{\ln x} + \frac{d}{x\sqrt{x}}$
- 2723 $y = a + bx^2 \ln x + c \frac{x}{\ln x} + d \frac{\ln x}{x^2}$
- 2724 $y = a + bx^2 \ln x + c \frac{x}{\ln x} + \frac{d}{x^2}$
- 2725 $y = a + bx^2 \ln x + c \frac{x}{\ln x} + d e^{-x}$
- 2726 $y = a + bx^2 \ln x + c\sqrt{x} + d \ln x$
- 2727 $y = a + bx^2 \ln x + c\sqrt{x} + \frac{d}{\ln x}$
- 2728 $y = a + bx^2 \ln x + c\sqrt{x} + \frac{d}{\sqrt{x}}$
- 2729 $y = a + bx^2 \ln x + c\sqrt{x} + d \frac{\ln x}{x}$
- 2730 $y = a + bx^2 \ln x + c\sqrt{x} + \frac{d}{x}$
- 2731 $y = a + bx^2 \ln x + c\sqrt{x} + \frac{d}{x\sqrt{x}}$
- 2732 $y = a + bx^2 \ln x + c\sqrt{x} + d \frac{\ln x}{x^2}$
- 2733 $y = a + bx^2 \ln x + c\sqrt{x} + \frac{d}{x^2}$
- 2734 $y = a + bx^2 \ln x + c\sqrt{x} + d e^{-x}$
- 2735 $y = a + bx^2 \ln x + c \ln x + \frac{d}{\ln x}$
- 2736 $y = a + bx^2 \ln x + c \ln x + \frac{d}{\sqrt{x}}$
- 2737 $y = a + bx^2 \ln x + c \ln x + d \frac{\ln x}{x}$
- 2738 $y = a + bx^2 \ln x + c \ln x + \frac{d}{x}$
- 2739 $y = a + bx^2 \ln x + c \ln x + \frac{d}{x\sqrt{x}}$
- 2740 $y = a + bx^2 \ln x + c \ln x + d \frac{\ln x}{x^2}$
- 2741 $y = a + bx^2 \ln x + c \ln x + \frac{d}{x^2}$
- 2742 $y = a + bx^2 \ln x + c \ln x + d e^{-x}$
- 2743 $y = a + bx^2 \ln x + \frac{c}{\ln x} + \frac{d}{\sqrt{x}}$
- 2744 $y = a + bx^2 \ln x + \frac{c}{\ln x} + d \frac{\ln x}{x}$
- 2745 $y = a + bx^2 \ln x + \frac{c}{\ln x} + \frac{d}{x}$
- 2746 $y = a + bx^2 \ln x + \frac{c}{\ln x} + \frac{d}{x\sqrt{x}}$
- 2747 $y = a + bx^2 \ln x + \frac{c}{\ln x} + d \frac{\ln x}{x^2}$
- 2748 $y = a + bx^2 \ln x + \frac{c}{\ln x} + \frac{d}{x^2}$
- 2749 $y = a + bx^2 \ln x + \frac{c}{\ln x} + d e^{-x}$
- 2750 $y = a + bx^2 \ln x + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x}$
- 2751 $y = a + bx^2 \ln x + \frac{c}{\sqrt{x}} + \frac{d}{x}$
- 2752 $y = a + bx^2 \ln x + \frac{c}{\sqrt{x}} + \frac{d}{x\sqrt{x}}$
- 2753 $y = a + bx^2 \ln x + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x^2}$
- 2754 $y = a + bx^2 \ln x + \frac{c}{\sqrt{x}} + \frac{d}{x^2}$
- 2755 $y = a + bx^2 \ln x + \frac{c}{\sqrt{x}} + d e^{-x}$
- 2756 $y = a + bx^2 \ln x + c \frac{\ln x}{x} + \frac{d}{x}$
- 2757 $y = a + bx^2 \ln x + c \frac{\ln x}{x} + \frac{d}{x\sqrt{x}}$
- 2758 $y = a + bx^2 \ln x + c \frac{\ln x}{x} + d \frac{\ln x}{x^2}$
- 2759 $y = a + bx^2 \ln x + c \frac{\ln x}{x} + \frac{d}{x^2}$
- 2760 $y = a + bx^2 \ln x + c \frac{\ln x}{x} + d e^{-x}$
- 2761 $y = a + bx^2 \ln x + \frac{c}{x} + \frac{d}{x\sqrt{x}}$
- 2762 $y = a + bx^2 \ln x + \frac{c}{x} + d \frac{\ln x}{x^2}$
- 2763 $y = a + bx^2 \ln x + \frac{c}{x} + \frac{d}{x^2}$
- 2764 $y = a + bx^2 \ln x + \frac{c}{x} + d e^{-x}$
- 2765 $y = a + bx^2 \ln x + \frac{c}{x\sqrt{x}} + d \frac{\ln x}{x^2}$
- 2766 $y = a + bx^2 \ln x + \frac{c}{x\sqrt{x}} + \frac{d}{x^2}$
- 2767 $y = a + bx^2 \ln x + \frac{c}{x\sqrt{x}} + d e^{-x}$
- 2768 $y = a + bx^2 \ln x + c \frac{\ln x}{x^2} + \frac{d}{x^2}$
- 2769 $y = a + bx^2 \ln x + c \frac{\ln x}{x^2} + d e^{-x}$
- 2770 $y = a + bx^2 \ln x + \frac{c}{x^2} + d e^{-x}$
- 2771 $y = a + bx^2 \sqrt{x} + cx^3 + d e^x$
- 2772 $y = a + bx^2 \sqrt{x} + cx^3 + d\sqrt{x} \ln x$
- 2773 $y = a + bx^2 \sqrt{x} + cx^3 + d(\ln x)^2$
- 2774 $y = a + bx^2 \sqrt{x} + cx^3 + d \frac{x}{\ln x}$

2775 $y = a + bx^2 \sqrt{x} + cx^3 + d\sqrt{x}$	2801 $y = a + bx^2 \sqrt{x} + c\sqrt{x} \ln x + d \ln x$	2827 $y = a + bx^2 \sqrt{x} + c \frac{x}{\ln x} + \frac{d}{x\sqrt{x}}$
2776 $y = a + bx^2 \sqrt{x} + cx^3 + d \ln x$	2802 $y = a + bx^2 \sqrt{x} + c\sqrt{x} \ln x + \frac{d}{\ln x}$	2828 $y = a + bx^2 \sqrt{x} + c \frac{x}{\ln x} + d \frac{\ln x}{x^2}$
2777 $y = a + bx^2 \sqrt{x} + cx^3 + \frac{d}{\ln x}$	2803 $y = a + bx^2 \sqrt{x} + c\sqrt{x} \ln x + \frac{d}{\sqrt{x}}$	2829 $y = a + bx^2 \sqrt{x} + c \frac{x}{\ln x} + \frac{d}{x^2}$
2778 $y = a + bx^2 \sqrt{x} + cx^3 + \frac{d}{\sqrt{x}}$	2804 $y = a + bx^2 \sqrt{x} + c\sqrt{x} \ln x + d \frac{\ln x}{x}$	2830 $y = a + bx^2 \sqrt{x} + c \frac{x}{\ln x} + d e^{-x}$
2779 $y = a + bx^2 \sqrt{x} + cx^3 + d \frac{\ln x}{x}$	2805 $y = a + bx^2 \sqrt{x} + c\sqrt{x} \ln x + \frac{d}{x}$	2831 $y = a + bx^2 \sqrt{x} + c\sqrt{x} + d \ln x$
2780 $y = a + bx^2 \sqrt{x} + cx^3 + \frac{d}{x}$	2806 $y = a + bx^2 \sqrt{x} + c\sqrt{x} \ln x + \frac{d}{x\sqrt{x}}$	2832 $y = a + bx^2 \sqrt{x} + c\sqrt{x} + \frac{d}{\ln x}$
2781 $y = a + bx^2 \sqrt{x} + cx^3 + \frac{d}{x\sqrt{x}}$	2807 $y = a + bx^2 \sqrt{x} + c\sqrt{x} \ln x + d \frac{\ln x}{x^2}$	2833 $y = a + bx^2 \sqrt{x} + c\sqrt{x} + \frac{d}{\sqrt{x}}$
2782 $y = a + bx^2 \sqrt{x} + cx^3 + d \frac{\ln x}{x^2}$	2808 $y = a + bx^2 \sqrt{x} + c\sqrt{x} \ln x + \frac{d}{x^2}$	2834 $y = a + bx^2 \sqrt{x} + c\sqrt{x} + d \frac{\ln x}{x}$
2783 $y = a + bx^2 \sqrt{x} + cx^3 + \frac{d}{x^2}$	2809 $y = a + bx^2 \sqrt{x} + c\sqrt{x} \ln x + d e^{-x}$	2835 $y = a + bx^2 \sqrt{x} + c\sqrt{x} + \frac{d}{x}$
2784 $y = a + bx^2 \sqrt{x} + cx^3 + d e^{-x}$	2810 $y = a + bx^2 \sqrt{x} + c(\ln x)^2 + d \frac{x}{\ln x}$	2836 $y = a + bx^2 \sqrt{x} + c\sqrt{x} + \frac{d}{x\sqrt{x}}$
2785 $y = a + bx^2 \sqrt{x} + c e^x + d\sqrt{x} \ln x$	2811 $y = a + bx^2 \sqrt{x} + c(\ln x)^2 + d\sqrt{x}$	2837 $y = a + bx^2 \sqrt{x} + c\sqrt{x} + d \frac{\ln x}{x^2}$
2786 $y = a + bx^2 \sqrt{x} + c e^x + d(\ln x)^2$	2812 $y = a + bx^2 \sqrt{x} + c(\ln x)^2 + d \ln x$	2838 $y = a + bx^2 \sqrt{x} + c\sqrt{x} + \frac{d}{x^2}$
2787 $y = a + bx^2 \sqrt{x} + c e^x + d \frac{x}{\ln x}$	2813 $y = a + bx^2 \sqrt{x} + c(\ln x)^2 + \frac{d}{\ln x}$	2839 $y = a + bx^2 \sqrt{x} + c\sqrt{x} + d e^{-x}$
2788 $y = a + bx^2 \sqrt{x} + c e^x + d\sqrt{x}$	2814 $y = a + bx^2 \sqrt{x} + c(\ln x)^2 + \frac{d}{\sqrt{x}}$	2840 $y = a + bx^2 \sqrt{x} + c \ln x + \frac{d}{\ln x}$
2789 $y = a + bx^2 \sqrt{x} + c e^x + d \ln x$	2815 $y = a + bx^2 \sqrt{x} + c(\ln x)^2 + d \frac{\ln x}{x}$	2841 $y = a + bx^2 \sqrt{x} + c \ln x + \frac{d}{\sqrt{x}}$
2790 $y = a + bx^2 \sqrt{x} + c e^x + \frac{d}{\ln x}$	2816 $y = a + bx^2 \sqrt{x} + c(\ln x)^2 + \frac{d}{x}$	2842 $y = a + bx^2 \sqrt{x} + c \ln x + d \frac{\ln x}{x}$
2791 $y = a + bx^2 \sqrt{x} + c e^x + \frac{d}{\sqrt{x}}$	2817 $y = a + bx^2 \sqrt{x} + c(\ln x)^2 + \frac{d}{x\sqrt{x}}$	2843 $y = a + bx^2 \sqrt{x} + c \ln x + \frac{d}{x}$
2792 $y = a + bx^2 \sqrt{x} + c e^x + d \frac{\ln x}{x}$	2818 $y = a + bx^2 \sqrt{x} + c(\ln x)^2 + d \frac{\ln x}{x^2}$	2844 $y = a + bx^2 \sqrt{x} + c \ln x + \frac{d}{x\sqrt{x}}$
2793 $y = a + bx^2 \sqrt{x} + c e^x + \frac{d}{x}$	2819 $y = a + bx^2 \sqrt{x} + c(\ln x)^2 + \frac{d}{x^2}$	2845 $y = a + bx^2 \sqrt{x} + c \ln x + d \frac{\ln x}{x^2}$
2794 $y = a + bx^2 \sqrt{x} + c e^x + \frac{d}{x\sqrt{x}}$	2820 $y = a + bx^2 \sqrt{x} + c(\ln x)^2 + d e^{-x}$	2846 $y = a + bx^2 \sqrt{x} + c \ln x + \frac{d}{x^2}$
2795 $y = a + bx^2 \sqrt{x} + c e^x + d \frac{\ln x}{x^2}$	2821 $y = a + bx^2 \sqrt{x} + c \frac{x}{\ln x} + d\sqrt{x}$	2847 $y = a + bx^2 \sqrt{x} + c \ln x + d e^{-x}$
2796 $y = a + bx^2 \sqrt{x} + c e^x + \frac{d}{x^2}$	2822 $y = a + bx^2 \sqrt{x} + c \frac{x}{\ln x} + d \ln x$	2848 $y = a + bx^2 \sqrt{x} + \frac{c}{\ln x} + \frac{d}{\sqrt{x}}$
2797 $y = a + bx^2 \sqrt{x} + c e^x + d e^{-x}$	2823 $y = a + bx^2 \sqrt{x} + c \frac{x}{\ln x} + \frac{d}{\ln x}$	2849 $y = a + bx^2 \sqrt{x} + \frac{c}{\ln x} + d \frac{\ln x}{x}$
2798 $y = a + bx^2 \sqrt{x} + c\sqrt{x} \ln x + d(\ln x)^2$	2824 $y = a + bx^2 \sqrt{x} + c \frac{x}{\ln x} + \frac{d}{\sqrt{x}}$	2850 $y = a + bx^2 \sqrt{x} + \frac{c}{\ln x} + \frac{d}{x}$
2799 $y = a + bx^2 \sqrt{x} + c\sqrt{x} \ln x + d \frac{x}{\ln x}$	2825 $y = a + bx^2 \sqrt{x} + c \frac{x}{\ln x} + d \frac{\ln x}{x}$	2851 $y = a + bx^2 \sqrt{x} + \frac{c}{\ln x} + \frac{d}{x\sqrt{x}}$
2800 $y = a + bx^2 \sqrt{x} + c\sqrt{x} \ln x + d\sqrt{x}$	2826 $y = a + bx^2 \sqrt{x} + c \frac{x}{\ln x} + \frac{d}{x}$	2852 $y = a + bx^2 \sqrt{x} + \frac{c}{\ln x} + d \frac{\ln x}{x^2}$

- 2853 $y = a + bx^2 \sqrt{x} + \frac{c}{\ln x} + \frac{d}{x^2}$
- 2854 $y = a + bx^2 \sqrt{x} + \frac{c}{\ln x} + d e^{-x}$
- 2855 $y = a + bx^2 \sqrt{x} + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x}$
- 2856 $y = a + bx^2 \sqrt{x} + \frac{c}{\sqrt{x}} + \frac{d}{x}$
- 2857 $y = a + bx^2 \sqrt{x} + \frac{c}{\sqrt{x}} + \frac{d}{x\sqrt{x}}$
- 2858 $y = a + bx^2 \sqrt{x} + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x^2}$
- 2859 $y = a + bx^2 \sqrt{x} + \frac{c}{\sqrt{x}} + \frac{d}{x^2}$
- 2860 $y = a + bx^2 \sqrt{x} + \frac{c}{\sqrt{x}} + d e^{-x}$
- 2861 $y = a + bx^2 \sqrt{x} + c \frac{\ln x}{x} + \frac{d}{x}$
- 2862 $y = a + bx^2 \sqrt{x} + c \frac{\ln x}{x} + \frac{d}{x\sqrt{x}}$
- 2863 $y = a + bx^2 \sqrt{x} + c \frac{\ln x}{x} + d \frac{\ln x}{x^2}$
- 2864 $y = a + bx^2 \sqrt{x} + c \frac{\ln x}{x} + \frac{d}{x^2}$
- 2865 $y = a + bx^2 \sqrt{x} + c \frac{\ln x}{x} + d e^{-x}$
- 2866 $y = a + bx^2 \sqrt{x} + \frac{c}{x} + \frac{d}{x\sqrt{x}}$
- 2867 $y = a + bx^2 \sqrt{x} + \frac{c}{x} + d \frac{\ln x}{x^2}$
- 2868 $y = a + bx^2 \sqrt{x} + \frac{c}{x} + \frac{d}{x^2}$
- 2869 $y = a + bx^2 \sqrt{x} + \frac{c}{x} + d e^{-x}$
- 2870 $y = a + bx^2 \sqrt{x} + \frac{c}{x\sqrt{x}} + d \frac{\ln x}{x^2}$
- 2871 $y = a + bx^2 \sqrt{x} + \frac{c}{x\sqrt{x}} + \frac{d}{x^2}$
- 2872 $y = a + bx^2 \sqrt{x} + \frac{c}{x\sqrt{x}} + d e^{-x}$
- 2873 $y = a + bx^2 \sqrt{x} + c \frac{\ln x}{x^2} + \frac{d}{x^2}$
- 2874 $y = a + bx^2 \sqrt{x} + c \frac{\ln x}{x^2} + d e^{-x}$
- 2875 $y = a + bx^2 \sqrt{x} + \frac{c}{x^2} + d e^{-x}$
- 2876 $y = a + bx^3 + c e^x + d \sqrt{x} \ln x$
- 2877 $y = a + bx^3 + c e^x + d (\ln x)^2$
- 2878 $y = a + bx^3 + c e^x + d \frac{x}{\ln x}$
- 2879 $y = a + bx^3 + c e^x + d \sqrt{x}$
- 2880 $y = a + bx^3 + c e^x + d \ln x$
- 2881 $y = a + bx^3 + c e^x + \frac{d}{\ln x}$
- 2882 $y = a + bx^3 + c e^x + \frac{d}{\sqrt{x}}$
- 2883 $y = a + bx^3 + c e^x + d \frac{\ln x}{x}$
- 2884 $y = a + bx^3 + c e^x + \frac{d}{x}$
- 2885 $y = a + bx^3 + c e^x + \frac{d}{x\sqrt{x}}$
- 2886 $y = a + bx^3 + c e^x + d \frac{\ln x}{x^2}$
- 2887 $y = a + bx^3 + c e^x + \frac{d}{x^2}$
- 2888 $y = a + bx^3 + c e^x + d e^{-x}$
- 2889 $y = a + bx^3 + c \sqrt{x} \ln x + d (\ln x)^2$
- 2890 $y = a + bx^3 + c \sqrt{x} \ln x + d \frac{x}{\ln x}$
- 2891 $y = a + bx^3 + c \sqrt{x} \ln x + d \sqrt{x}$
- 2892 $y = a + bx^3 + c \sqrt{x} \ln x + d \ln x$
- 2893 $y = a + bx^3 + c \sqrt{x} \ln x + \frac{d}{\ln x}$
- 2894 $y = a + bx^3 + c \sqrt{x} \ln x + \frac{d}{\sqrt{x}}$
- 2895 $y = a + bx^3 + c \sqrt{x} \ln x + d \frac{\ln x}{x}$
- 2896 $y = a + bx^3 + c \sqrt{x} \ln x + \frac{d}{x}$
- 2897 $y = a + bx^3 + c \sqrt{x} \ln x + \frac{d}{x\sqrt{x}}$
- 2898 $y = a + bx^3 + c \sqrt{x} \ln x + d \frac{\ln x}{x^2}$
- 2899 $y = a + bx^3 + c \sqrt{x} \ln x + \frac{d}{x^2}$
- 2900 $y = a + bx^3 + c \sqrt{x} \ln x + d e^{-x}$
- 2901 $y = a + bx^3 + c (\ln x)^2 + d \frac{x}{\ln x}$
- 2902 $y = a + bx^3 + c (\ln x)^2 + d \sqrt{x}$
- 2903 $y = a + bx^3 + c (\ln x)^2 + d \ln x$
- 2904 $y = a + bx^3 + c (\ln x)^2 + \frac{d}{\ln x}$
- 2905 $y = a + bx^3 + c (\ln x)^2 + \frac{d}{\sqrt{x}}$
- 2906 $y = a + bx^3 + c (\ln x)^2 + d \frac{\ln x}{x}$
- 2907 $y = a + bx^3 + c (\ln x)^2 + \frac{d}{x}$
- 2908 $y = a + bx^3 + c (\ln x)^2 + \frac{d}{x\sqrt{x}}$
- 2909 $y = a + bx^3 + c (\ln x)^2 + d \frac{\ln x}{x^2}$
- 2910 $y = a + bx^3 + c (\ln x)^2 + \frac{d}{x^2}$
- 2911 $y = a + bx^3 + c (\ln x)^2 + d e^{-x}$
- 2912 $y = a + bx^3 + c \frac{x}{\ln x} + d \sqrt{x}$
- 2913 $y = a + bx^3 + c \frac{x}{\ln x} + d \ln x$
- 2914 $y = a + bx^3 + c \frac{x}{\ln x} + \frac{d}{\ln x}$
- 2915 $y = a + bx^3 + c \frac{x}{\ln x} + \frac{d}{\sqrt{x}}$
- 2916 $y = a + bx^3 + c \frac{x}{\ln x} + d \frac{\ln x}{x}$
- 2917 $y = a + bx^3 + c \frac{x}{\ln x} + \frac{d}{x}$
- 2918 $y = a + bx^3 + c \frac{x}{\ln x} + \frac{d}{x\sqrt{x}}$
- 2919 $y = a + bx^3 + c \frac{x}{\ln x} + d \frac{\ln x}{x^2}$
- 2920 $y = a + bx^3 + c \frac{x}{\ln x} + \frac{d}{x^2}$
- 2921 $y = a + bx^3 + c \frac{x}{\ln x} + d e^{-x}$
- 2922 $y = a + bx^3 + c \sqrt{x} + d \ln x$
- 2923 $y = a + bx^3 + c \sqrt{x} + \frac{d}{\ln x}$
- 2924 $y = a + bx^3 + c \sqrt{x} + \frac{d}{\sqrt{x}}$
- 2925 $y = a + bx^3 + c \sqrt{x} + d \frac{\ln x}{x}$
- 2926 $y = a + bx^3 + c \sqrt{x} + \frac{d}{x}$
- 2927 $y = a + bx^3 + c \sqrt{x} + \frac{d}{x\sqrt{x}}$
- 2928 $y = a + bx^3 + c \sqrt{x} + d \frac{\ln x}{x^2}$
- 2929 $y = a + bx^3 + c \sqrt{x} + \frac{d}{x^2}$
- 2930 $y = a + bx^3 + c \sqrt{x} + d e^{-x}$

- 2931 $y = a + bx^3 + c \ln x + \frac{d}{\ln x}$
- 2932 $y = a + bx^3 + c \ln x + \frac{d}{\sqrt{x}}$
- 2933 $y = a + bx^3 + c \ln x + d \frac{\ln x}{x}$
- 2934 $y = a + bx^3 + c \ln x + \frac{d}{x}$
- 2935 $y = a + bx^3 + c \ln x + \frac{d}{x\sqrt{x}}$
- 2936 $y = a + bx^3 + c \ln x + d \frac{\ln x}{x^2}$
- 2937 $y = a + bx^3 + c \ln x + \frac{d}{x^2}$
- 2938 $y = a + bx^3 + c \ln x + d e^{-x}$
- 2939 $y = a + bx^3 + \frac{c}{\ln x} + \frac{d}{\sqrt{x}}$
- 2940 $y = a + bx^3 + \frac{c}{\ln x} + d \frac{\ln x}{x}$
- 2941 $y = a + bx^3 + \frac{c}{\ln x} + \frac{d}{x}$
- 2942 $y = a + bx^3 + \frac{c}{\ln x} + \frac{d}{x\sqrt{x}}$
- 2943 $y = a + bx^3 + \frac{c}{\ln x} + d \frac{\ln x}{x^2}$
- 2944 $y = a + bx^3 + \frac{c}{\ln x} + \frac{d}{x^2}$
- 2945 $y = a + bx^3 + \frac{c}{\ln x} + d e^{-x}$
- 2946 $y = a + bx^3 + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x}$
- 2947 $y = a + bx^3 + \frac{c}{\sqrt{x}} + \frac{d}{x}$
- 2948 $y = a + bx^3 + \frac{c}{\sqrt{x}} + \frac{d}{x\sqrt{x}}$
- 2949 $y = a + bx^3 + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x^2}$
- 2950 $y = a + bx^3 + \frac{c}{\sqrt{x}} + \frac{d}{x^2}$
- 2951 $y = a + bx^3 + \frac{c}{\sqrt{x}} + d e^{-x}$
- 2952 $y = a + bx^3 + c \frac{\ln x}{x} + \frac{d}{x}$
- 2953 $y = a + bx^3 + c \frac{\ln x}{x} + \frac{d}{x\sqrt{x}}$
- 2954 $y = a + bx^3 + c \frac{\ln x}{x} + d \frac{\ln x}{x^2}$
- 2955 $y = a + bx^3 + c \frac{\ln x}{x} + \frac{d}{x^2}$
- 2956 $y = a + bx^3 + c \frac{\ln x}{x} + d e^{-x}$
- 2957 $y = a + bx^3 + \frac{c}{x} + \frac{d}{x\sqrt{x}}$
- 2958 $y = a + bx^3 + \frac{c}{x} + d \frac{\ln x}{x^2}$
- 2959 $y = a + bx^3 + \frac{c}{x} + \frac{d}{x^2}$
- 2960 $y = a + bx^3 + \frac{c}{x} + d e^{-x}$
- 2961 $y = a + bx^3 + \frac{c}{x\sqrt{x}} + d \frac{\ln x}{x^2}$
- 2962 $y = a + bx^3 + \frac{c}{x\sqrt{x}} + \frac{d}{x^2}$
- 2963 $y = a + bx^3 + \frac{c}{x\sqrt{x}} + d e^{-x}$
- 2964 $y = a + bx^3 + c \frac{\ln x}{x^2} + \frac{d}{x^2}$
- 2965 $y = a + bx^3 + c \frac{\ln x}{x^2} + d e^{-x}$
- 2966 $y = a + bx^3 + \frac{c}{x^2} + d e^{-x}$
- 2967 $y = a + b e^x + c\sqrt{x} \ln x + d(\ln x)^2$
- 2968 $y = a + b e^x + c\sqrt{x} \ln x + d \frac{x}{\ln x}$
- 2969 $y = a + b e^x + c\sqrt{x} \ln x + d\sqrt{x}$
- 2970 $y = a + b e^x + c\sqrt{x} \ln x + d \ln x$
- 2971 $y = a + b e^x + c\sqrt{x} \ln x + \frac{d}{\ln x}$
- 2972 $y = a + b e^x + c\sqrt{x} \ln x + \frac{d}{\sqrt{x}}$
- 2973 $y = a + b e^x + c\sqrt{x} \ln x + d \frac{\ln x}{x}$
- 2974 $y = a + b e^x + c\sqrt{x} \ln x + \frac{d}{x}$
- 2975 $y = a + b e^x + c\sqrt{x} \ln x + \frac{d}{x\sqrt{x}}$
- 2976 $y = a + b e^x + c\sqrt{x} \ln x + d \frac{\ln x}{x^2}$
- 2977 $y = a + b e^x + c\sqrt{x} \ln x + \frac{d}{x^2}$
- 2978 $y = a + b e^x + c\sqrt{x} \ln x + d e^{-x}$
- 2979 $y = a + b e^x + c(\ln x)^2 + d \frac{x}{\ln x}$
- 2980 $y = a + b e^x + c(\ln x)^2 + d\sqrt{x}$
- 2981 $y = a + b e^x + c(\ln x)^2 + d \ln x$
- 2982 $y = a + b e^x + c(\ln x)^2 + \frac{d}{\ln x}$
- 2983 $y = a + b e^x + c(\ln x)^2 + \frac{d}{\sqrt{x}}$
- 2984 $y = a + b e^x + c(\ln x)^2 + d \frac{\ln x}{x}$
- 2985 $y = a + b e^x + c(\ln x)^2 + \frac{d}{x}$
- 2986 $y = a + b e^x + c(\ln x)^2 + \frac{d}{x\sqrt{x}}$
- 2987 $y = a + b e^x + c(\ln x)^2 + d \frac{\ln x}{x^2}$
- 2988 $y = a + b e^x + c(\ln x)^2 + \frac{d}{x^2}$
- 2989 $y = a + b e^x + c(\ln x)^2 + d e^{-x}$
- 2990 $y = a + b e^x + c \frac{x}{\ln x} + d\sqrt{x}$
- 2991 $y = a + b e^x + c \frac{x}{\ln x} + d \ln x$
- 2992 $y = a + b e^x + c \frac{x}{\ln x} + \frac{d}{\ln x}$
- 2993 $y = a + b e^x + c \frac{x}{\ln x} + \frac{d}{\sqrt{x}}$
- 2994 $y = a + b e^x + c \frac{x}{\ln x} + d \frac{\ln x}{x}$
- 2995 $y = a + b e^x + c \frac{x}{\ln x} + \frac{d}{x}$
- 2996 $y = a + b e^x + c \frac{x}{\ln x} + \frac{d}{x\sqrt{x}}$
- 2997 $y = a + b e^x + c \frac{x}{\ln x} + d \frac{\ln x}{x^2}$
- 2998 $y = a + b e^x + c \frac{x}{\ln x} + \frac{d}{x^2}$
- 2999 $y = a + b e^x + c \frac{x}{\ln x} + d e^{-x}$
- 3000 $y = a + b e^x + c\sqrt{x} + d \ln x$
- 3001 $y = a + b e^x + c\sqrt{x} + \frac{d}{\ln x}$
- 3002 $y = a + b e^x + c\sqrt{x} + \frac{d}{\sqrt{x}}$
- 3003 $y = a + b e^x + c\sqrt{x} + d \frac{\ln x}{x}$
- 3004 $y = a + b e^x + c\sqrt{x} + \frac{d}{x}$
- 3005 $y = a + b e^x + c\sqrt{x} + \frac{d}{x\sqrt{x}}$
- 3006 $y = a + b e^x + c\sqrt{x} + d \frac{\ln x}{x^2}$
- 3007 $y = a + b e^x + c\sqrt{x} + \frac{d}{x^2}$
- 3008 $y = a + b e^x + c\sqrt{x} + d e^{-x}$

- 3009 $y = a + b e^x + c \ln x + \frac{d}{\ln x}$
- 3010 $y = a + b e^x + c \ln x + \frac{d}{\sqrt{x}}$
- 3011 $y = a + b e^x + c \ln x + d \frac{\ln x}{x}$
- 3012 $y = a + b e^x + c \ln x + \frac{d}{x}$
- 3013 $y = a + b e^x + c \ln x + \frac{d}{x\sqrt{x}}$
- 3014 $y = a + b e^x + c \ln x + d \frac{\ln x}{x^2}$
- 3015 $y = a + b e^x + c \ln x + \frac{d}{x^2}$
- 3016 $y = a + b e^x + c \ln x + d e^{-x}$
- 3017 $y = a + b e^x + \frac{c}{\ln x} + \frac{d}{\sqrt{x}}$
- 3018 $y = a + b e^x + \frac{c}{\ln x} + d \frac{\ln x}{x}$
- 3019 $y = a + b e^x + \frac{c}{\ln x} + \frac{d}{x}$
- 3020 $y = a + b e^x + \frac{c}{\ln x} + \frac{d}{x\sqrt{x}}$
- 3021 $y = a + b e^x + \frac{c}{\ln x} + d \frac{\ln x}{x^2}$
- 3022 $y = a + b e^x + \frac{c}{\ln x} + \frac{d}{x^2}$
- 3023 $y = a + b e^x + \frac{c}{\ln x} + d e^{-x}$
- 3024 $y = a + b e^x + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x}$
- 3025 $y = a + b e^x + \frac{c}{\sqrt{x}} + \frac{d}{x}$
- 3026 $y = a + b e^x + \frac{c}{\sqrt{x}} + \frac{d}{x\sqrt{x}}$
- 3027 $y = a + b e^x + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x^2}$
- 3028 $y = a + b e^x + \frac{c}{\sqrt{x}} + \frac{d}{x^2}$
- 3029 $y = a + b e^x + \frac{c}{\sqrt{x}} + d e^{-x}$
- 3030 $y = a + b e^x + c \frac{\ln x}{x} + \frac{d}{x}$
- 3031 $y = a + b e^x + c \frac{\ln x}{x} + \frac{d}{x\sqrt{x}}$
- 3032 $y = a + b e^x + c \frac{\ln x}{x} + d \frac{\ln x}{x^2}$
- 3033 $y = a + b e^x + c \frac{\ln x}{x} + \frac{d}{x^2}$
- 3034 $y = a + b e^x + c \frac{\ln x}{x} + d e^{-x}$
- 3035 $y = a + b e^x + \frac{c}{x} + \frac{d}{x\sqrt{x}}$
- 3036 $y = a + b e^x + \frac{c}{x} + d \frac{\ln x}{x^2}$
- 3037 $y = a + b e^x + \frac{c}{x} + \frac{d}{x^2}$
- 3038 $y = a + b e^x + \frac{c}{x} + d e^{-x}$
- 3039 $y = a + b e^x + \frac{c}{x\sqrt{x}} + d \frac{\ln x}{x^2}$
- 3040 $y = a + b e^x + \frac{c}{x\sqrt{x}} + \frac{d}{x^2}$
- 3041 $y = a + b e^x + \frac{c}{x\sqrt{x}} + d e^{-x}$
- 3042 $y = a + b e^x + c \frac{\ln x}{x^2} + \frac{d}{x^2}$
- 3043 $y = a + b e^x + c \frac{\ln x}{x^2} + d e^{-x}$
- 3044 $y = a + b e^x + \frac{c}{x^2} + d e^{-x}$
- 3045 $y = a + b\sqrt{x} \ln x + c(\ln x)^2 + d \frac{x}{\ln x}$
- 3046 $y = a + b\sqrt{x} \ln x + c(\ln x)^2 + d\sqrt{x}$
- 3047 $y = a + b\sqrt{x} \ln x + c(\ln x)^2 + d \ln x$
- 3048 $y = a + b\sqrt{x} \ln x + c(\ln x)^2 + \frac{d}{\ln x}$
- 3049 $y = a + b\sqrt{x} \ln x + c(\ln x)^2 + \frac{d}{\sqrt{x}}$
- 3050 $y = a + b\sqrt{x} \ln x + c(\ln x)^2 + d \frac{\ln x}{x}$
- 3051 $y = a + b\sqrt{x} \ln x + c(\ln x)^2 + \frac{d}{x}$
- 3052 $y = a + b\sqrt{x} \ln x + c(\ln x)^2 + \frac{d}{x\sqrt{x}}$
- 3053 $y = a + b\sqrt{x} \ln x + c(\ln x)^2 + d \frac{\ln x}{x^2}$
- 3054 $y = a + b\sqrt{x} \ln x + c(\ln x)^2 + \frac{d}{x^2}$
- 3055 $y = a + b\sqrt{x} \ln x + c(\ln x)^2 + d e^{-x}$
- 3056 $y = a + b\sqrt{x} \ln x + c \frac{x}{\ln x} + d\sqrt{x}$
- 3057 $y = a + b\sqrt{x} \ln x + c \frac{x}{\ln x} + d \ln x$
- 3058 $y = a + b\sqrt{x} \ln x + c \frac{x}{\ln x} + \frac{d}{\ln x}$
- 3059 $y = a + b\sqrt{x} \ln x + c \frac{x}{\ln x} + \frac{d}{\sqrt{x}}$
- 3060 $y = a + b\sqrt{x} \ln x + c \frac{x}{\ln x} + d \frac{\ln x}{x}$
- 3061 $y = a + b\sqrt{x} \ln x + c \frac{x}{\ln x} + \frac{d}{x}$
- 3062 $y = a + b\sqrt{x} \ln x + c \frac{x}{\ln x} + \frac{d}{x\sqrt{x}}$
- 3063 $y = a + b\sqrt{x} \ln x + c \frac{x}{\ln x} + d \frac{\ln x}{x^2}$
- 3064 $y = a + b\sqrt{x} \ln x + c \frac{x}{\ln x} + \frac{d}{x^2}$
- 3065 $y = a + b\sqrt{x} \ln x + c \frac{x}{\ln x} + d e^{-x}$
- 3066 $y = a + b\sqrt{x} \ln x + c\sqrt{x} + d \ln x$
- 3067 $y = a + b\sqrt{x} \ln x + c\sqrt{x} + \frac{d}{\ln x}$
- 3068 $y = a + b\sqrt{x} \ln x + c\sqrt{x} + \frac{d}{\sqrt{x}}$
- 3069 $y = a + b\sqrt{x} \ln x + c\sqrt{x} + d \frac{\ln x}{x}$
- 3070 $y = a + b\sqrt{x} \ln x + c\sqrt{x} + \frac{d}{x}$
- 3071 $y = a + b\sqrt{x} \ln x + c\sqrt{x} + \frac{d}{x\sqrt{x}}$
- 3072 $y = a + b\sqrt{x} \ln x + c\sqrt{x} + d \frac{\ln x}{x^2}$
- 3073 $y = a + b\sqrt{x} \ln x + c\sqrt{x} + \frac{d}{x^2}$
- 3074 $y = a + b\sqrt{x} \ln x + c\sqrt{x} + d e^{-x}$
- 3075 $y = a + b\sqrt{x} \ln x + c \ln x + \frac{d}{\ln x}$
- 3076 $y = a + b\sqrt{x} \ln x + c \ln x + \frac{d}{\sqrt{x}}$
- 3077 $y = a + b\sqrt{x} \ln x + c \ln x + d \frac{\ln x}{x}$
- 3078 $y = a + b\sqrt{x} \ln x + c \ln x + \frac{d}{x}$
- 3079 $y = a + b\sqrt{x} \ln x + c \ln x + \frac{d}{x\sqrt{x}}$
- 3080 $y = a + b\sqrt{x} \ln x + c \ln x + d \frac{\ln x}{x^2}$
- 3081 $y = a + b\sqrt{x} \ln x + c \ln x + \frac{d}{x^2}$
- 3082 $y = a + b\sqrt{x} \ln x + c \ln x + d e^{-x}$
- 3083 $y = a + b\sqrt{x} \ln x + \frac{c}{\ln x} + \frac{d}{\sqrt{x}}$
- 3084 $y = a + b\sqrt{x} \ln x + \frac{c}{\ln x} + d \frac{\ln x}{x}$
- 3085 $y = a + b\sqrt{x} \ln x + \frac{c}{\ln x} + \frac{d}{x}$
- 3086 $y = a + b\sqrt{x} \ln x + \frac{c}{\ln x} + \frac{d}{x\sqrt{x}}$

$$\begin{aligned}
 3087 \quad y &= a + b\sqrt{x} \ln x + \frac{c}{\ln x} + d \frac{\ln x}{x^2} \\
 3088 \quad y &= a + b\sqrt{x} \ln x + \frac{c}{\ln x} + \frac{d}{x^2} \\
 3089 \quad y &= a + b\sqrt{x} \ln x + \frac{c}{\ln x} + d e^{-x} \\
 3090 \quad y &= a + b\sqrt{x} \ln x + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x} \\
 3091 \quad y &= a + b\sqrt{x} \ln x + \frac{c}{\sqrt{x}} + \frac{d}{x} \\
 3092 \quad y &= a + b\sqrt{x} \ln x + \frac{c}{\sqrt{x}} + \frac{d}{x\sqrt{x}} \\
 3093 \quad y &= a + b\sqrt{x} \ln x + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x^2} \\
 3094 \quad y &= a + b\sqrt{x} \ln x + \frac{c}{\sqrt{x}} + \frac{d}{x^2} \\
 3095 \quad y &= a + b\sqrt{x} \ln x + \frac{c}{\sqrt{x}} + d e^{-x} \\
 3096 \quad y &= a + b\sqrt{x} \ln x + c \frac{\ln x}{x} + \frac{d}{x} \\
 3097 \quad y &= a + b\sqrt{x} \ln x + c \frac{\ln x}{x} + \frac{d}{x\sqrt{x}} \\
 3098 \quad y &= a + b\sqrt{x} \ln x + c \frac{\ln x}{x} + d \frac{\ln x}{x^2} \\
 3099 \quad y &= a + b\sqrt{x} \ln x + c \frac{\ln x}{x} + \frac{d}{x^2} \\
 3100 \quad y &= a + b\sqrt{x} \ln x + c \frac{\ln x}{x} + d e^{-x} \\
 3101 \quad y &= a + b\sqrt{x} \ln x + \frac{c}{x} + \frac{d}{x\sqrt{x}} \\
 3102 \quad y &= a + b\sqrt{x} \ln x + \frac{c}{x} + d \frac{\ln x}{x^2} \\
 3103 \quad y &= a + b\sqrt{x} \ln x + \frac{c}{x} + \frac{d}{x^2} \\
 3104 \quad y &= a + b\sqrt{x} \ln x + \frac{c}{x} + d e^{-x} \\
 3105 \quad y &= a + b\sqrt{x} \ln x + \frac{c}{x\sqrt{x}} + d \frac{\ln x}{x^2} \\
 3106 \quad y &= a + b\sqrt{x} \ln x + \frac{c}{x\sqrt{x}} + \frac{d}{x^2} \\
 3107 \quad y &= a + b\sqrt{x} \ln x + \frac{c}{x\sqrt{x}} + d e^{-x} \\
 3108 \quad y &= a + b\sqrt{x} \ln x + c \frac{\ln x}{x^2} + \frac{d}{x^2} \\
 3109 \quad y &= a + b\sqrt{x} \ln x + c \frac{\ln x}{x^2} + d e^{-x} \\
 3110 \quad y &= a + b\sqrt{x} \ln x + \frac{c}{x^2} + d e^{-x} \\
 3111 \quad y &= a + b(\ln x)^2 + c \frac{x}{\ln x} + d\sqrt{x} \\
 3112 \quad y &= a + b(\ln x)^2 + c \frac{x}{\ln x} + d \ln x
 \end{aligned}$$

$$\begin{aligned}
 3113 \quad y &= a + b(\ln x)^2 + c \frac{x}{\ln x} + \frac{d}{\ln x} \\
 3114 \quad y &= a + b(\ln x)^2 + c \frac{x}{\ln x} + \frac{d}{\sqrt{x}} \\
 3115 \quad y &= a + b(\ln x)^2 + c \frac{x}{\ln x} + d \frac{\ln x}{x} \\
 3116 \quad y &= a + b(\ln x)^2 + c \frac{x}{\ln x} + \frac{d}{x} \\
 3117 \quad y &= a + b(\ln x)^2 + c \frac{x}{\ln x} + \frac{d}{x\sqrt{x}} \\
 3118 \quad y &= a + b(\ln x)^2 + c \frac{x}{\ln x} + d \frac{\ln x}{x^2} \\
 3119 \quad y &= a + b(\ln x)^2 + c \frac{x}{\ln x} + \frac{d}{x^2} \\
 3120 \quad y &= a + b(\ln x)^2 + c \frac{x}{\ln x} + d e^{-x} \\
 3121 \quad y &= a + b(\ln x)^2 + c\sqrt{x} + d \ln x \\
 3122 \quad y &= a + b(\ln x)^2 + c\sqrt{x} + \frac{d}{\ln x} \\
 3123 \quad y &= a + b(\ln x)^2 + c\sqrt{x} + \frac{d}{\sqrt{x}} \\
 3124 \quad y &= a + b(\ln x)^2 + c\sqrt{x} + d \frac{\ln x}{x} \\
 3125 \quad y &= a + b(\ln x)^2 + c\sqrt{x} + \frac{d}{x} \\
 3126 \quad y &= a + b(\ln x)^2 + c\sqrt{x} + \frac{d}{x\sqrt{x}} \\
 3127 \quad y &= a + b(\ln x)^2 + c\sqrt{x} + d \frac{\ln x}{x^2} \\
 3128 \quad y &= a + b(\ln x)^2 + c\sqrt{x} + \frac{d}{x^2} \\
 3129 \quad y &= a + b(\ln x)^2 + c\sqrt{x} + d e^{-x} \\
 3130 \quad y &= a + b(\ln x)^2 + c \ln x + \frac{d}{\ln x} \\
 3131 \quad y &= a + b(\ln x)^2 + c \ln x + \frac{d}{\sqrt{x}} \\
 3132 \quad y &= a + b(\ln x)^2 + c \ln x + d \frac{\ln x}{x} \\
 3133 \quad y &= a + b(\ln x)^2 + c \ln x + \frac{d}{x} \\
 3134 \quad y &= a + b(\ln x)^2 + c \ln x + \frac{d}{x\sqrt{x}} \\
 3135 \quad y &= a + b(\ln x)^2 + c \ln x + d \frac{\ln x}{x^2} \\
 3136 \quad y &= a + b(\ln x)^2 + c \ln x + \frac{d}{x^2} \\
 3137 \quad y &= a + b(\ln x)^2 + c \ln x + d e^{-x} \\
 3138 \quad y &= a + b(\ln x)^2 + \frac{c}{\ln x} + \frac{d}{\sqrt{x}}
 \end{aligned}$$

$$\begin{aligned}
 3139 \quad y &= a + b(\ln x)^2 + \frac{c}{\ln x} + d \frac{\ln x}{x} \\
 3140 \quad y &= a + b(\ln x)^2 + \frac{c}{\ln x} + \frac{d}{x} \\
 3141 \quad y &= a + b(\ln x)^2 + \frac{c}{\ln x} + \frac{d}{x\sqrt{x}} \\
 3142 \quad y &= a + b(\ln x)^2 + \frac{c}{\ln x} + d \frac{\ln x}{x^2} \\
 3143 \quad y &= a + b(\ln x)^2 + \frac{c}{\ln x} + \frac{d}{x^2} \\
 3144 \quad y &= a + b(\ln x)^2 + \frac{c}{\ln x} + d e^{-x} \\
 3145 \quad y &= a + b(\ln x)^2 + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x} \\
 3146 \quad y &= a + b(\ln x)^2 + \frac{c}{\sqrt{x}} + \frac{d}{x} \\
 3147 \quad y &= a + b(\ln x)^2 + \frac{c}{\sqrt{x}} + \frac{d}{x\sqrt{x}} \\
 3148 \quad y &= a + b(\ln x)^2 + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x^2} \\
 3149 \quad y &= a + b(\ln x)^2 + \frac{c}{\sqrt{x}} + \frac{d}{x^2} \\
 3150 \quad y &= a + b(\ln x)^2 + \frac{c}{\sqrt{x}} + d e^{-x} \\
 3151 \quad y &= a + b(\ln x)^2 + c \frac{\ln x}{x} + \frac{d}{x} \\
 3152 \quad y &= a + b(\ln x)^2 + c \frac{\ln x}{x} + \frac{d}{x\sqrt{x}} \\
 3153 \quad y &= a + b(\ln x)^2 + c \frac{\ln x}{x} + d \frac{\ln x}{x^2} \\
 3154 \quad y &= a + b(\ln x)^2 + c \frac{\ln x}{x} + \frac{d}{x^2} \\
 3155 \quad y &= a + b(\ln x)^2 + c \frac{\ln x}{x} + d e^{-x} \\
 3156 \quad y &= a + b(\ln x)^2 + \frac{c}{x} + \frac{d}{x\sqrt{x}} \\
 3157 \quad y &= a + b(\ln x)^2 + \frac{c}{x} + d \frac{\ln x}{x^2} \\
 3158 \quad y &= a + b(\ln x)^2 + \frac{c}{x} + \frac{d}{x^2} \\
 3159 \quad y &= a + b(\ln x)^2 + \frac{c}{x} + d e^{-x} \\
 3160 \quad y &= a + b(\ln x)^2 + \frac{c}{x\sqrt{x}} + d \frac{\ln x}{x^2} \\
 3161 \quad y &= a + b(\ln x)^2 + \frac{c}{x\sqrt{x}} + \frac{d}{x^2} \\
 3162 \quad y &= a + b(\ln x)^2 + \frac{c}{x\sqrt{x}} + d e^{-x} \\
 3163 \quad y &= a + b(\ln x)^2 + c \frac{\ln x}{x^2} + \frac{d}{x^2} \\
 3164 \quad y &= a + b(\ln x)^2 + c \frac{\ln x}{x^2} + d e^{-x}
 \end{aligned}$$

$$3165 \ y = a + b(\ln x)^2 + \frac{c}{x^2} + d e^{-x}$$

$$3166 \ y = a + b \frac{x}{\ln x} + c\sqrt{x} + d \ln x$$

$$3167 \ y = a + b \frac{x}{\ln x} + c\sqrt{x} + \frac{d}{\ln x}$$

$$3168 \ y = a + b \frac{x}{\ln x} + c\sqrt{x} + \frac{d}{\sqrt{x}}$$

$$3169 \ y = a + b \frac{x}{\ln x} + c\sqrt{x} + d \frac{\ln x}{x}$$

$$3170 \ y = a + b \frac{x}{\ln x} + c\sqrt{x} + \frac{d}{x}$$

$$3171 \ y = a + b \frac{x}{\ln x} + c\sqrt{x} + \frac{d}{x\sqrt{x}}$$

$$3172 \ y = a + b \frac{x}{\ln x} + c\sqrt{x} + d \frac{\ln x}{x^2}$$

$$3173 \ y = a + b \frac{x}{\ln x} + c\sqrt{x} + \frac{d}{x^2}$$

$$3174 \ y = a + b \frac{x}{\ln x} + c\sqrt{x} + d e^{-x}$$

$$3175 \ y = a + b \frac{x}{\ln x} + c \ln x + \frac{d}{\ln x}$$

$$3176 \ y = a + b \frac{x}{\ln x} + c \ln x + \frac{d}{\sqrt{x}}$$

$$3177 \ y = a + b \frac{x}{\ln x} + c \ln x + d \frac{\ln x}{x}$$

$$3178 \ y = a + b \frac{x}{\ln x} + c \ln x + \frac{d}{x}$$

$$3179 \ y = a + b \frac{x}{\ln x} + c \ln x + \frac{d}{x\sqrt{x}}$$

$$3180 \ y = a + b \frac{x}{\ln x} + c \ln x + d \frac{\ln x}{x^2}$$

$$3181 \ y = a + b \frac{x}{\ln x} + c \ln x + \frac{d}{x^2}$$

$$3182 \ y = a + b \frac{x}{\ln x} + c \ln x + d e^{-x}$$

$$3183 \ y = a + b \frac{x}{\ln x} + \frac{c}{\ln x} + \frac{d}{\sqrt{x}}$$

$$3184 \ y = a + b \frac{x}{\ln x} + \frac{c}{\ln x} + d \frac{\ln x}{x}$$

$$3185 \ y = a + b \frac{x}{\ln x} + \frac{c}{\ln x} + \frac{d}{x}$$

$$3186 \ y = a + b \frac{x}{\ln x} + \frac{c}{\ln x} + \frac{d}{x\sqrt{x}}$$

$$3187 \ y = a + b \frac{x}{\ln x} + \frac{c}{\ln x} + d \frac{\ln x}{x^2}$$

$$3188 \ y = a + b \frac{x}{\ln x} + \frac{c}{\ln x} + \frac{d}{x^2}$$

$$3189 \ y = a + b \frac{x}{\ln x} + \frac{c}{\ln x} + d e^{-x}$$

$$3190 \ y = a + b \frac{x}{\ln x} + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x}$$

$$3191 \ y = a + b \frac{x}{\ln x} + \frac{c}{\sqrt{x}} + \frac{d}{x}$$

$$3192 \ y = a + b \frac{x}{\ln x} + \frac{c}{\sqrt{x}} + \frac{d}{x\sqrt{x}}$$

$$3193 \ y = a + b \frac{x}{\ln x} + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x^2}$$

$$3194 \ y = a + b \frac{x}{\ln x} + \frac{c}{\sqrt{x}} + \frac{d}{x^2}$$

$$3195 \ y = a + b \frac{x}{\ln x} + \frac{c}{\sqrt{x}} + d e^{-x}$$

$$3196 \ y = a + b \frac{x}{\ln x} + c \frac{\ln x}{x} + \frac{d}{x}$$

$$3197 \ y = a + b \frac{x}{\ln x} + c \frac{\ln x}{x} + \frac{d}{x\sqrt{x}}$$

$$3198 \ y = a + b \frac{x}{\ln x} + c \frac{\ln x}{x} + d \frac{\ln x}{x^2}$$

$$3199 \ y = a + b \frac{x}{\ln x} + c \frac{\ln x}{x} + \frac{d}{x^2}$$

$$3200 \ y = a + b \frac{x}{\ln x} + c \frac{\ln x}{x} + d e^{-x}$$

$$3201 \ y = a + b \frac{x}{\ln x} + \frac{c}{x} + \frac{d}{x\sqrt{x}}$$

$$3202 \ y = a + b \frac{x}{\ln x} + \frac{c}{x} + d \frac{\ln x}{x^2}$$

$$3203 \ y = a + b \frac{x}{\ln x} + \frac{c}{x} + \frac{d}{x^2}$$

$$3204 \ y = a + b \frac{x}{\ln x} + \frac{c}{x} + d e^{-x}$$

$$3205 \ y = a + b \frac{x}{\ln x} + \frac{c}{x\sqrt{x}} + d \frac{\ln x}{x^2}$$

$$3206 \ y = a + b \frac{x}{\ln x} + \frac{c}{x\sqrt{x}} + \frac{d}{x^2}$$

$$3207 \ y = a + b \frac{x}{\ln x} + \frac{c}{x\sqrt{x}} + d e^{-x}$$

$$3208 \ y = a + b \frac{x}{\ln x} + c \frac{\ln x}{x^2} + \frac{d}{x^2}$$

$$3209 \ y = a + b \frac{x}{\ln x} + c \frac{\ln x}{x^2} + d e^{-x}$$

$$3210 \ y = a + b \frac{x}{\ln x} + \frac{c}{x^2} + d e^{-x}$$

$$3211 \ y = a + b\sqrt{x} + c \ln x + \frac{d}{\ln x}$$

$$3212 \ y = a + b\sqrt{x} + c \ln x + \frac{d}{\sqrt{x}}$$

$$3213 \ y = a + b\sqrt{x} + c \ln x + d \frac{\ln x}{x}$$

$$3214 \ y = a + b\sqrt{x} + c \ln x + \frac{d}{x}$$

$$3215 \ y = a + b\sqrt{x} + c \ln x + \frac{d}{x\sqrt{x}}$$

$$3216 \ y = a + b\sqrt{x} + c \ln x + d \frac{\ln x}{x^2}$$

$$3217 \ y = a + b\sqrt{x} + c \ln x + \frac{d}{x^2}$$

$$3218 \ y = a + b\sqrt{x} + c \ln x + d e^{-x}$$

$$3219 \ y = a + b\sqrt{x} + \frac{c}{\ln x} + \frac{d}{\sqrt{x}}$$

$$3220 \ y = a + b\sqrt{x} + \frac{c}{\ln x} + d \frac{\ln x}{x}$$

$$3221 \ y = a + b\sqrt{x} + \frac{c}{\ln x} + \frac{d}{x}$$

$$3222 \ y = a + b\sqrt{x} + \frac{c}{\ln x} + \frac{d}{x\sqrt{x}}$$

$$3223 \ y = a + b\sqrt{x} + \frac{c}{\ln x} + d \frac{\ln x}{x^2}$$

$$3224 \ y = a + b\sqrt{x} + \frac{c}{\ln x} + \frac{d}{x^2}$$

$$3225 \ y = a + b\sqrt{x} + \frac{c}{\ln x} + d e^{-x}$$

$$3226 \ y = a + b\sqrt{x} + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x}$$

$$3227 \ y = a + b\sqrt{x} + \frac{c}{\sqrt{x}} + \frac{d}{x}$$

$$3228 \ y = a + b\sqrt{x} + \frac{c}{\sqrt{x}} + \frac{d}{x\sqrt{x}}$$

$$3229 \ y = a + b\sqrt{x} + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x^2}$$

$$3230 \ y = a + b\sqrt{x} + \frac{c}{\sqrt{x}} + \frac{d}{x^2}$$

$$3231 \ y = a + b\sqrt{x} + \frac{c}{\sqrt{x}} + d e^{-x}$$

$$3232 \ y = a + b\sqrt{x} + c \frac{\ln x}{x} + \frac{d}{x}$$

$$3233 \ y = a + b\sqrt{x} + c \frac{\ln x}{x} + \frac{d}{x\sqrt{x}}$$

$$3234 \ y = a + b\sqrt{x} + c \frac{\ln x}{x} + d \frac{\ln x}{x^2}$$

$$3235 \ y = a + b\sqrt{x} + c \frac{\ln x}{x} + \frac{d}{x^2}$$

$$3236 \ y = a + b\sqrt{x} + c \frac{\ln x}{x} + d e^{-x}$$

$$3237 \ y = a + b\sqrt{x} + \frac{c}{x} + \frac{d}{x\sqrt{x}}$$

$$3238 \ y = a + b\sqrt{x} + \frac{c}{x} + d \frac{\ln x}{x^2}$$

$$3239 \ y = a + b\sqrt{x} + \frac{c}{x} + \frac{d}{x^2}$$

$$3240 \ y = a + b\sqrt{x} + \frac{c}{x} + d e^{-x}$$

$$3241 \ y = a + b\sqrt{x} + \frac{c}{x\sqrt{x}} + d \frac{\ln x}{x^2}$$

$$3242 \ y = a + b\sqrt{x} + \frac{c}{x\sqrt{x}} + \frac{d}{x^2}$$

$$3243 \ y = a + b\sqrt{x} + \frac{c}{x\sqrt{x}} + d e^{-x}$$

$$3244 \ y = a + b\sqrt{x} + c \frac{\ln x}{x^2} + \frac{d}{x^2}$$

$$3245 \ y = a + b\sqrt{x} + c \frac{\ln x}{x^2} + d e^{-x}$$

$$3246 \ y = a + b\sqrt{x} + \frac{c}{x^2} + d e^{-x}$$

$$3247 \ y = a + b \ln x + \frac{c}{\ln x} + \frac{d}{\sqrt{x}}$$

$$3248 \ y = a + b \ln x + \frac{c}{\ln x} + d \frac{\ln x}{x}$$

$$3249 \ y = a + b \ln x + \frac{c}{\ln x} + \frac{d}{x}$$

$$3250 \ y = a + b \ln x + \frac{c}{\ln x} + \frac{d}{x\sqrt{x}}$$

$$3251 \ y = a + b \ln x + \frac{c}{\ln x} + d \frac{\ln x}{x^2}$$

$$3252 \ y = a + b \ln x + \frac{c}{\ln x} + \frac{d}{x^2}$$

$$3253 \ y = a + b \ln x + \frac{c}{\ln x} + d e^{-x}$$

$$3254 \ y = a + b \ln x + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x}$$

$$3255 \ y = a + b \ln x + \frac{c}{\sqrt{x}} + \frac{d}{x}$$

$$3256 \ y = a + b \ln x + \frac{c}{\sqrt{x}} + \frac{d}{x\sqrt{x}}$$

$$3257 \ y = a + b \ln x + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x^2}$$

$$3258 \ y = a + b \ln x + \frac{c}{\sqrt{x}} + \frac{d}{x^2}$$

$$3259 \ y = a + b \ln x + \frac{c}{\sqrt{x}} + d e^{-x}$$

$$3260 \ y = a + b \ln x + c \frac{\ln x}{x} + \frac{d}{x}$$

$$3261 \ y = a + b \ln x + c \frac{\ln x}{x} + \frac{d}{x\sqrt{x}}$$

$$3262 \ y = a + b \ln x + c \frac{\ln x}{x} + d \frac{\ln x}{x^2}$$

$$3263 \ y = a + b \ln x + c \frac{\ln x}{x} + \frac{d}{x^2}$$

$$3264 \ y = a + b \ln x + c \frac{\ln x}{x} + d e^{-x}$$

$$3265 \ y = a + b \ln x + \frac{c}{x} + \frac{d}{x\sqrt{x}}$$

$$3266 \ y = a + b \ln x + \frac{c}{x} + d \frac{\ln x}{x^2}$$

$$3267 \ y = a + b \ln x + \frac{c}{x} + \frac{d}{x^2}$$

$$3268 \ y = a + b \ln x + \frac{c}{x} + d e^{-x}$$

$$3269 \ y = a + b \ln x + \frac{c}{x\sqrt{x}} + d \frac{\ln x}{x^2}$$

$$3270 \ y = a + b \ln x + \frac{c}{x\sqrt{x}} + \frac{d}{x^2}$$

$$3271 \ y = a + b \ln x + \frac{c}{x\sqrt{x}} + d e^{-x}$$

$$3272 \ y = a + b \ln x + c \frac{\ln x}{x^2} + \frac{d}{x^2}$$

$$3273 \ y = a + b \ln x + c \frac{\ln x}{x^2} + d e^{-x}$$

$$3274 \ y = a + b \ln x + \frac{c}{x^2} + d e^{-x}$$

$$3275 \ y = a + \frac{b}{\ln x} + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x}$$

$$3276 \ y = a + \frac{b}{\ln x} + \frac{c}{\sqrt{x}} + \frac{d}{x}$$

$$3277 \ y = a + \frac{b}{\ln x} + \frac{c}{\sqrt{x}} + \frac{d}{x\sqrt{x}}$$

$$3278 \ y = a + \frac{b}{\ln x} + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x^2}$$

$$3279 \ y = a + \frac{b}{\ln x} + \frac{c}{\sqrt{x}} + \frac{d}{x^2}$$

$$3280 \ y = a + \frac{b}{\ln x} + \frac{c}{\sqrt{x}} + d e^{-x}$$

$$3281 \ y = a + \frac{b}{\ln x} + c \frac{\ln x}{x} + \frac{d}{x}$$

$$3282 \ y = a + \frac{b}{\ln x} + c \frac{\ln x}{x} + \frac{d}{x\sqrt{x}}$$

$$3283 \ y = a + \frac{b}{\ln x} + c \frac{\ln x}{x} + d \frac{\ln x}{x^2}$$

$$3284 \ y = a + \frac{b}{\ln x} + c \frac{\ln x}{x} + \frac{d}{x^2}$$

$$3285 \ y = a + \frac{b}{\ln x} + c \frac{\ln x}{x} + d e^{-x}$$

$$3286 \ y = a + \frac{b}{\ln x} + \frac{c}{x} + \frac{d}{x\sqrt{x}}$$

$$3287 \ y = a + \frac{b}{\ln x} + \frac{c}{x} + d \frac{\ln x}{x^2}$$

$$3288 \ y = a + \frac{b}{\ln x} + \frac{c}{x} + \frac{d}{x^2}$$

$$3289 \ y = a + \frac{b}{\ln x} + \frac{c}{x} + d e^{-x}$$

$$3290 \ y = a + \frac{b}{\ln x} + \frac{c}{x\sqrt{x}} + d \frac{\ln x}{x^2}$$

$$3291 \ y = a + \frac{b}{\ln x} + \frac{c}{x\sqrt{x}} + \frac{d}{x^2}$$

$$3292 \ y = a + \frac{b}{\ln x} + \frac{c}{x\sqrt{x}} + d e^{-x}$$

$$3293 \ y = a + \frac{b}{\ln x} + c \frac{\ln x}{x^2} + \frac{d}{x^2}$$

$$3294 \ y = a + \frac{b}{\ln x} + c \frac{\ln x}{x^2} + d e^{-x}$$

$$3295 \ y = a + \frac{b}{\ln x} + \frac{c}{x^2} + d e^{-x}$$

$$3296 \ y = a + \frac{b}{\sqrt{x}} + c \frac{\ln x}{x} + \frac{d}{x}$$

$$3297 \ y = a + \frac{b}{\sqrt{x}} + c \frac{\ln x}{x} + \frac{d}{x\sqrt{x}}$$

$$3298 \ y = a + \frac{b}{\sqrt{x}} + c \frac{\ln x}{x} + d \frac{\ln x}{x^2}$$

$$3299 \ y = a + \frac{b}{\sqrt{x}} + c \frac{\ln x}{x} + \frac{d}{x^2}$$

$$3300 \ y = a + \frac{b}{\sqrt{x}} + c \frac{\ln x}{x} + d e^{-x}$$

$$3301 \ y = a + \frac{b}{\sqrt{x}} + \frac{c}{x} + \frac{d}{x\sqrt{x}}$$

$$3302 \ y = a + \frac{b}{\sqrt{x}} + \frac{c}{x} + d \frac{\ln x}{x^2}$$

$$3303 \ y = a + \frac{b}{\sqrt{x}} + \frac{c}{x} + \frac{d}{x^2}$$

$$3304 \ y = a + \frac{b}{\sqrt{x}} + \frac{c}{x} + d e^{-x}$$

$$3305 \ y = a + \frac{b}{\sqrt{x}} + \frac{c}{x\sqrt{x}} + d \frac{\ln x}{x^2}$$

$$3306 \ y = a + \frac{b}{\sqrt{x}} + \frac{c}{x\sqrt{x}} + \frac{d}{x^2}$$

$$3307 \ y = a + \frac{b}{\sqrt{x}} + \frac{c}{x\sqrt{x}} + d e^{-x}$$

$$3308 \ y = a + \frac{b}{\sqrt{x}} + c \frac{\ln x}{x^2} + \frac{d}{x^2}$$

$$3309 \ y = a + \frac{b}{\sqrt{x}} + c \frac{\ln x}{x^2} + d e^{-x}$$

$$3310 \ y = a + \frac{b}{\sqrt{x}} + \frac{c}{x^2} + d e^{-x}$$

$$3311 \ y = a + b \frac{\ln x}{x} + \frac{c}{x} + \frac{d}{x\sqrt{x}}$$

$$3312 \ y = a + b \frac{\ln x}{x} + \frac{c}{x} + d \frac{\ln x}{x^2}$$

$$3313 \ y = a + b \frac{\ln x}{x} + \frac{c}{x} + \frac{d}{x^2}$$

$$3314 \ y = a + b \frac{\ln x}{x} + \frac{c}{x} + d e^{-x}$$

$$3315 \ y = a + b \frac{\ln x}{x} + \frac{c}{x\sqrt{x}} + d \frac{\ln x}{x^2}$$

$$3316 \ y = a + b \frac{\ln x}{x} + \frac{c}{x\sqrt{x}} + \frac{d}{x^2}$$

$$3317 \ y = a + b \frac{\ln x}{x} + \frac{c}{x\sqrt{x}} + d e^{-x}$$

$$3318 \ y = a + b \frac{\ln x}{x} + c \frac{\ln x}{x^2} + \frac{d}{x^2}$$

$$3319 \ y = a + b \frac{\ln x}{x} + c \frac{\ln x}{x^2} + d e^{-x}$$

$$3320 \ y = a + b \frac{\ln x}{x} + \frac{c}{x^2} + d e^{-x}$$

$$3321 \ y = a + \frac{b}{x} + \frac{c}{x\sqrt{x}} + d \frac{\ln x}{x^2}$$

$$3322 \ y = a + \frac{b}{x} + \frac{c}{x\sqrt{x}} + \frac{d}{x^2}$$

$$3323 \ y = a + \frac{b}{x} + \frac{c}{x\sqrt{x}} + d e^{-x}$$

$$3324 \ y = a + \frac{b}{x} + c \frac{\ln x}{x^2} + \frac{d}{x^2}$$

$$3325 \ y = a + \frac{b}{x} + c \frac{\ln x}{x^2} + d e^{-x}$$

$$3326 \ y = a + \frac{b}{x} + \frac{c}{x^2} + d e^{-x}$$

$$3327 \ y = a + \frac{b}{x\sqrt{x}} + c \frac{\ln x}{x^2} + \frac{d}{x^2}$$

$$3328 \ y = a + \frac{b}{x\sqrt{x}} + c \frac{\ln x}{x^2} + d e^{-x}$$

$$3329 \ y = a + \frac{b}{x\sqrt{x}} + \frac{c}{x^2} + d e^{-x}$$

$$3330 \ y = a + b \frac{\ln x}{x^2} + \frac{c}{x^2} + d e^{-x}$$

Eqns 4001-5140: Four Term Functions with Intercept

4001 $y = a + bx + cx \ln x + dx\sqrt{x} + ex^2$	4025 $y = a + bx + cx \ln x + dx^2 + e\frac{x}{\ln x}$	4049 $y = a + bx + cx \ln x + dx^2 \ln x + e\frac{\ln x}{x^2}$
4002 $y = a + bx + cx \ln x + dx\sqrt{x} + ex^2 \ln x$	4026 $y = a + bx + cx \ln x + dx^2 + e\sqrt{x}$	4050 $y = a + bx + cx \ln x + dx^2 \ln x + \frac{e}{x^2}$
4003 $y = a + bx + cx \ln x + dx\sqrt{x} + ex^2 \sqrt{x}$	4027 $y = a + bx + cx \ln x + dx^2 + e \ln x$	4051 $y = a + bx + cx \ln x + dx^2 \ln x + e e^{-x}$
4004 $y = a + bx + cx \ln x + dx\sqrt{x} + ex^3$	4028 $y = a + bx + cx \ln x + dx^2 + \frac{e}{\ln x}$	4052 $y = a + bx + cx \ln x + dx^2 \sqrt{x} + ex^3$
4005 $y = a + bx + cx \ln x + dx\sqrt{x} + e e^x$	4029 $y = a + bx + cx \ln x + dx^2 + e\frac{e}{\sqrt{x}}$	4053 $y = a + bx + cx \ln x + dx^2 \sqrt{x} + e e^x$
4006 $y = a + bx + cx \ln x + dx\sqrt{x} + e\sqrt{x} \ln x$	4030 $y = a + bx + cx \ln x + dx^2 + e\frac{\ln x}{x}$	4054 $y = a + bx + cx \ln x + dx^2 \sqrt{x} + e\sqrt{x} \ln x$
4007 $y = a + bx + cx \ln x + dx\sqrt{x} + e(\ln x)^2$	4031 $y = a + bx + cx \ln x + dx^2 + \frac{e}{x}$	4055 $y = a + bx + cx \ln x + dx^2 \sqrt{x} + e(\ln x)^2$
4008 $y = a + bx + cx \ln x + dx\sqrt{x} + e\frac{x}{\ln x}$	4032 $y = a + bx + cx \ln x + dx^2 + \frac{e}{x\sqrt{x}}$	4056 $y = a + bx + cx \ln x + dx^2 \sqrt{x} + e\frac{x}{\ln x}$
4009 $y = a + bx + cx \ln x + dx\sqrt{x} + e\sqrt{x}$	4033 $y = a + bx + cx \ln x + dx^2 + e\frac{\ln x}{x^2}$	4057 $y = a + bx + cx \ln x + dx^2 \sqrt{x} + e\sqrt{x}$
4010 $y = a + bx + cx \ln x + dx\sqrt{x} + e \ln x$	4034 $y = a + bx + cx \ln x + dx^2 + \frac{e}{x^2}$	4058 $y = a + bx + cx \ln x + dx^2 \sqrt{x} + e \ln x$
4011 $y = a + bx + cx \ln x + dx\sqrt{x} + \frac{e}{\ln x}$	4035 $y = a + bx + cx \ln x + dx^2 + e e^{-x}$	4059 $y = a + bx + cx \ln x + dx^2 \sqrt{x} + \frac{e}{\ln x}$
4012 $y = a + bx + cx \ln x + dx\sqrt{x} + \frac{e}{\sqrt{x}}$	4036 $y = a + bx + cx \ln x + dx^2 \ln x + ex^2 \sqrt{x}$	4060 $y = a + bx + cx \ln x + dx^2 \sqrt{x} + \frac{e}{\sqrt{x}}$
4013 $y = a + bx + cx \ln x + dx\sqrt{x} + e\frac{\ln x}{x}$	4037 $y = a + bx + cx \ln x + dx^2 \ln x + ex^3$	4061 $y = a + bx + cx \ln x + dx^2 \sqrt{x} + e\frac{\ln x}{x}$
4014 $y = a + bx + cx \ln x + dx\sqrt{x} + \frac{e}{x}$	4038 $y = a + bx + cx \ln x + dx^2 \ln x + e e^x$	4062 $y = a + bx + cx \ln x + dx^2 \sqrt{x} + \frac{e}{x}$
4015 $y = a + bx + cx \ln x + dx\sqrt{x} + \frac{e}{x\sqrt{x}}$	4039 $y = a + bx + cx \ln x + dx^2 \ln x + e\sqrt{x} \ln x$	4063 $y = a + bx + cx \ln x + dx^2 \sqrt{x} + \frac{e}{x\sqrt{x}}$
4016 $y = a + bx + cx \ln x + dx\sqrt{x} + e\frac{\ln x}{x^2}$	4040 $y = a + bx + cx \ln x + dx^2 \ln x + e(\ln x)^2$	4064 $y = a + bx + cx \ln x + dx^2 \sqrt{x} + e\frac{\ln x}{x^2}$
4017 $y = a + bx + cx \ln x + dx\sqrt{x} + \frac{e}{x^2}$	4041 $y = a + bx + cx \ln x + dx^2 \ln x + e\frac{x}{\ln x}$	4065 $y = a + bx + cx \ln x + dx^2 \sqrt{x} + \frac{e}{x^2}$
4018 $y = a + bx + cx \ln x + dx\sqrt{x} + e e^{-x}$	4042 $y = a + bx + cx \ln x + dx^2 \ln x + e\sqrt{x}$	4066 $y = a + bx + cx \ln x + dx^2 \sqrt{x} + e e^{-x}$
4019 $y = a + bx + cx \ln x + dx^2 + ex^2 \ln x$	4043 $y = a + bx + cx \ln x + dx^2 \ln x + e \ln x$	4067 $y = a + bx + cx \ln x + dx^3 + e e^x$
4020 $y = a + bx + cx \ln x + dx^2 + ex^2 \sqrt{x}$	4044 $y = a + bx + cx \ln x + dx^2 \ln x + \frac{e}{\ln x}$	4068 $y = a + bx + cx \ln x + dx^3 + e\sqrt{x} \ln x$
4021 $y = a + bx + cx \ln x + dx^2 + ex^3$	4045 $y = a + bx + cx \ln x + dx^2 \ln x + \frac{e}{\sqrt{x}}$	4069 $y = a + bx + cx \ln x + dx^3 + e(\ln x)^2$
4022 $y = a + bx + cx \ln x + dx^2 + e e^x$	4046 $y = a + bx + cx \ln x + dx^2 \ln x + e\frac{\ln x}{x}$	4070 $y = a + bx + cx \ln x + dx^3 + e\frac{x}{\ln x}$
4023 $y = a + bx + cx \ln x + dx^2 + e\sqrt{x} \ln x$	4047 $y = a + bx + cx \ln x + dx^2 \ln x + \frac{e}{x}$	4071 $y = a + bx + cx \ln x + dx^3 + e\sqrt{x}$
4024 $y = a + bx + cx \ln x + dx^2 + e(\ln x)^2$	4048 $y = a + bx + cx \ln x + dx^2 \ln x + \frac{e}{x\sqrt{x}}$	4072 $y = a + bx + cx \ln x + dx^3 + e \ln x$

- 4073 $y = a + bx + cx \ln x + dx^3 + \frac{e}{\ln x}$
- 4074 $y = a + bx + cx \ln x + dx^3 + \frac{e}{\sqrt{x}}$
- 4075 $y = a + bx + cx \ln x + dx^3 + e \frac{\ln x}{x}$
- 4076 $y = a + bx + cx \ln x + dx^3 + \frac{e}{x}$
- 4077 $y = a + bx + cx \ln x + dx^3 + \frac{e}{x\sqrt{x}}$
- 4078 $y = a + bx + cx \ln x + dx^3 + e \frac{\ln x}{x^2}$
- 4079 $y = a + bx + cx \ln x + dx^3 + \frac{e}{x^2}$
- 4080 $y = a + bx + cx \ln x + dx^3 + e e^{-x}$
- 4081 $y = a + bx + cx \ln x + d e^x + e\sqrt{x} \ln x$
- 4082 $y = a + bx + cx \ln x + d e^x + e(\ln x)^2$
- 4083 $y = a + bx + cx \ln x + d e^x + e \frac{x}{\ln x}$
- 4084 $y = a + bx + cx \ln x + d e^x + e\sqrt{x}$
- 4085 $y = a + bx + cx \ln x + d e^x + e \ln x$
- 4086 $y = a + bx + cx \ln x + d e^x + \frac{e}{\ln x}$
- 4087 $y = a + bx + cx \ln x + d e^x + \frac{e}{\sqrt{x}}$
- 4088 $y = a + bx + cx \ln x + d e^x + e \frac{\ln x}{x}$
- 4089 $y = a + bx + cx \ln x + d e^x + \frac{e}{x}$
- 4090 $y = a + bx + cx \ln x + d e^x + \frac{e}{x\sqrt{x}}$
- 4091 $y = a + bx + cx \ln x + d e^x + e \frac{\ln x}{x^2}$
- 4092 $y = a + bx + cx \ln x + d e^x + \frac{e}{x^2}$
- 4093 $y = a + bx + cx \ln x + d e^x + e e^{-x}$
- 4094 $y = a + bx + cx \ln x + d\sqrt{x} \ln x + e(\ln x)^2$
- 4095 $y = a + bx + cx \ln x + d\sqrt{x} \ln x + e \frac{x}{\ln x}$
- 4096 $y = a + bx + cx \ln x + d\sqrt{x} \ln x + e\sqrt{x}$
- 4097 $y = a + bx + cx \ln x + d\sqrt{x} \ln x + e \ln x$
- 4098 $y = a + bx + cx \ln x + d\sqrt{x} \ln x + \frac{e}{\ln x}$
- 4099 $y = a + bx + cx \ln x + d\sqrt{x} \ln x + \frac{e}{\sqrt{x}}$
- 4100 $y = a + bx + cx \ln x + d\sqrt{x} \ln x + e \frac{\ln x}{x}$
- 4101 $y = a + bx + cx \ln x + d\sqrt{x} \ln x + \frac{e}{x}$
- 4102 $y = a + bx + cx \ln x + d\sqrt{x} \ln x + \frac{e}{x\sqrt{x}}$
- 4103 $y = a + bx + cx \ln x + d\sqrt{x} \ln x + e \frac{\ln x}{x^2}$
- 4104 $y = a + bx + cx \ln x + d\sqrt{x} \ln x + \frac{e}{x^2}$
- 4105 $y = a + bx + cx \ln x + d\sqrt{x} \ln x + e e^{-x}$
- 4106 $y = a + bx + cx \ln x + d(\ln x)^2 + e \frac{x}{\ln x}$
- 4107 $y = a + bx + cx \ln x + d(\ln x)^2 + e\sqrt{x}$
- 4108 $y = a + bx + cx \ln x + d(\ln x)^2 + e \ln x$
- 4109 $y = a + bx + cx \ln x + d(\ln x)^2 + \frac{e}{\ln x}$
- 4110 $y = a + bx + cx \ln x + d(\ln x)^2 + \frac{e}{\sqrt{x}}$
- 4111 $y = a + bx + cx \ln x + d(\ln x)^2 + e \frac{\ln x}{x}$
- 4112 $y = a + bx + cx \ln x + d(\ln x)^2 + \frac{e}{x}$
- 4113 $y = a + bx + cx \ln x + d(\ln x)^2 + \frac{e}{x\sqrt{x}}$
- 4114 $y = a + bx + cx \ln x + d(\ln x)^2 + e \frac{\ln x}{x^2}$
- 4115 $y = a + bx + cx \ln x + d(\ln x)^2 + \frac{e}{x^2}$
- 4116 $y = a + bx + cx \ln x + d(\ln x)^2 + e e^{-x}$
- 4117 $y = a + bx + cx \ln x + d \frac{x}{\ln x} + e\sqrt{x}$
- 4118 $y = a + bx + cx \ln x + d \frac{x}{\ln x} + e \ln x$
- 4119 $y = a + bx + cx \ln x + d \frac{x}{\ln x} + \frac{e}{\ln x}$
- 4120 $y = a + bx + cx \ln x + d \frac{x}{\ln x} + \frac{e}{\sqrt{x}}$
- 4121 $y = a + bx + cx \ln x + d \frac{x}{\ln x} + e \frac{\ln x}{x}$
- 4122 $y = a + bx + cx \ln x + d \frac{x}{\ln x} + \frac{e}{x}$
- 4123 $y = a + bx + cx \ln x + d \frac{x}{\ln x} + \frac{e}{x\sqrt{x}}$
- 4124 $y = a + bx + cx \ln x + d \frac{x}{\ln x} + e \frac{\ln x}{x^2}$
- 4125 $y = a + bx + cx \ln x + d \frac{x}{\ln x} + \frac{e}{x^2}$
- 4126 $y = a + bx + cx \ln x + d \frac{x}{\ln x} + e e^{-x}$
- 4127 $y = a + bx + cx \ln x + d\sqrt{x} + e \ln x$
- 4128 $y = a + bx + cx \ln x + d\sqrt{x} + \frac{e}{\ln x}$
- 4129 $y = a + bx + cx \ln x + d\sqrt{x} + \frac{e}{\sqrt{x}}$
- 4130 $y = a + bx + cx \ln x + d\sqrt{x} + e \frac{\ln x}{x}$
- 4131 $y = a + bx + cx \ln x + d\sqrt{x} + \frac{e}{x}$
- 4132 $y = a + bx + cx \ln x + d\sqrt{x} + \frac{e}{x\sqrt{x}}$
- 4133 $y = a + bx + cx \ln x + d\sqrt{x} + e \frac{\ln x}{x^2}$
- 4134 $y = a + bx + cx \ln x + d\sqrt{x} + \frac{e}{x^2}$
- 4135 $y = a + bx + cx \ln x + d\sqrt{x} + e e^{-x}$
- 4136 $y = a + bx + cx \ln x + d \ln x + \frac{e}{\ln x}$
- 4137 $y = a + bx + cx \ln x + d \ln x + \frac{e}{\sqrt{x}}$
- 4138 $y = a + bx + cx \ln x + d \ln x + e \frac{\ln x}{x}$
- 4139 $y = a + bx + cx \ln x + d \ln x + \frac{e}{x}$
- 4140 $y = a + bx + cx \ln x + d \ln x + \frac{e}{x\sqrt{x}}$
- 4141 $y = a + bx + cx \ln x + d \ln x + e \frac{\ln x}{x^2}$
- 4142 $y = a + bx + cx \ln x + d \ln x + \frac{e}{x^2}$
- 4143 $y = a + bx + cx \ln x + d \ln x + e e^{-x}$
- 4144 $y = a + bx + cx \ln x + \frac{d}{\ln x} + \frac{e}{\sqrt{x}}$
- 4145 $y = a + bx + cx \ln x + \frac{d}{\ln x} + e \frac{\ln x}{x}$
- 4146 $y = a + bx + cx \ln x + \frac{d}{\ln x} + \frac{e}{x}$
- 4147 $y = a + bx + cx \ln x + \frac{d}{\ln x} + \frac{e}{x\sqrt{x}}$
- 4148 $y = a + bx + cx \ln x + \frac{d}{\ln x} + e \frac{\ln x}{x^2}$
- 4149 $y = a + bx + cx \ln x + \frac{d}{\ln x} + \frac{e}{x^2}$
- 4150 $y = a + bx + cx \ln x + \frac{d}{\ln x} + e e^{-x}$

$$4151 \ y = a + bx + cx \ln x + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x}$$

$$4152 \ y = a + bx + cx \ln x + \frac{d}{\sqrt{x}} + \frac{e}{x}$$

$$4153 \ y = a + bx + cx \ln x + \frac{d}{\sqrt{x}} + \frac{e}{x\sqrt{x}}$$

$$4154 \ y = a + bx + cx \ln x + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x^2}$$

$$4155 \ y = a + bx + cx \ln x + \frac{d}{\sqrt{x}} + \frac{e}{x^2}$$

$$4156 \ y = a + bx + cx \ln x + \frac{d}{\sqrt{x}} + e e^{-x}$$

$$4157 \ y = a + bx + cx \ln x + d \frac{\ln x}{x} + \frac{e}{x}$$

$$4158 \ y = a + bx + cx \ln x + d \frac{\ln x}{x} + \frac{e}{x\sqrt{x}}$$

$$4159 \ y = a + bx + cx \ln x + d \frac{\ln x}{x} + e \frac{\ln x}{x^2}$$

$$4160 \ y = a + bx + cx \ln x + d \frac{\ln x}{x} + \frac{e}{x^2}$$

$$4161 \ y = a + bx + cx \ln x + d \frac{\ln x}{x} + e e^{-x}$$

$$4162 \ y = a + bx + cx \ln x + \frac{d}{x} + \frac{e}{x\sqrt{x}}$$

$$4163 \ y = a + bx + cx \ln x + \frac{d}{x} + e \frac{\ln x}{x^2}$$

$$4164 \ y = a + bx + cx \ln x + \frac{d}{x} + \frac{e}{x^2}$$

$$4165 \ y = a + bx + cx \ln x + \frac{d}{x} + e e^{-x}$$

$$4166 \ y = a + bx + cx \ln x + \frac{d}{x\sqrt{x}} + e \frac{\ln x}{x^2}$$

$$4167 \ y = a + bx + cx \ln x + \frac{d}{x\sqrt{x}} + \frac{e}{x^2}$$

$$4168 \ y = a + bx + cx \ln x + \frac{d}{x\sqrt{x}} + e e^{-x}$$

$$4169 \ y = a + bx + cx \ln x + d \frac{\ln x}{x^2} + \frac{e}{x^2}$$

$$4170 \ y = a + bx + cx \ln x + d \frac{\ln x}{x^2} + e e^{-x}$$

$$4171 \ y = a + bx + cx \ln x + \frac{d}{x^2} + e e^{-x}$$

$$4172 \ y = a + bx + cx\sqrt{x} + dx^2 + ex^2 \ln x$$

$$4173 \ y = a + bx + cx\sqrt{x} + dx^2 + ex^2 \sqrt{x}$$

$$4174 \ y = a + bx + cx\sqrt{x} + dx^2 + ex^3$$

$$4175 \ y = a + bx + cx\sqrt{x} + dx^2 + e e^x$$

$$4176 \ y = a + bx + cx\sqrt{x} + dx^2 + e\sqrt{x} \ln x$$

$$4177 \ y = a + bx + cx\sqrt{x} + dx^2 + e(\ln x)^2$$

$$4178 \ y = a + bx + cx\sqrt{x} + dx^2 + e \frac{x}{\ln x}$$

$$4179 \ y = a + bx + cx\sqrt{x} + dx^2 + e\sqrt{x}$$

$$4180 \ y = a + bx + cx\sqrt{x} + dx^2 + e \ln x$$

$$4181 \ y = a + bx + cx\sqrt{x} + dx^2 + \frac{e}{\ln x}$$

$$4182 \ y = a + bx + cx\sqrt{x} + dx^2 + \frac{e}{\sqrt{x}}$$

$$4183 \ y = a + bx + cx\sqrt{x} + dx^2 + e \frac{\ln x}{x}$$

$$4184 \ y = a + bx + cx\sqrt{x} + dx^2 + \frac{e}{x}$$

$$4185 \ y = a + bx + cx\sqrt{x} + dx^2 + \frac{e}{x\sqrt{x}}$$

$$4186 \ y = a + bx + cx\sqrt{x} + dx^2 + e \frac{\ln x}{x^2}$$

$$4187 \ y = a + bx + cx\sqrt{x} + dx^2 + \frac{e}{x^2}$$

$$4188 \ y = a + bx + cx\sqrt{x} + dx^2 + e e^{-x}$$

$$4189 \ y = a + bx + cx\sqrt{x} + dx^2 \ln x + ex^2 \sqrt{x}$$

$$4190 \ y = a + bx + cx\sqrt{x} + dx^2 \ln x + ex^3$$

$$4191 \ y = a + bx + cx\sqrt{x} + dx^2 \ln x + e e^x$$

$$4192 \ y = a + bx + cx\sqrt{x} + dx^2 \ln x + e\sqrt{x} \ln x$$

$$4193 \ y = a + bx + cx\sqrt{x} + dx^2 \ln x + e(\ln x)^2$$

$$4194 \ y = a + bx + cx\sqrt{x} + dx^2 \ln x + e \frac{x}{\ln x}$$

$$4195 \ y = a + bx + cx\sqrt{x} + dx^2 \ln x + e\sqrt{x}$$

$$4196 \ y = a + bx + cx\sqrt{x} + dx^2 \ln x + e \ln x$$

$$4197 \ y = a + bx + cx\sqrt{x} + dx^2 \ln x + \frac{e}{\ln x}$$

$$4198 \ y = a + bx + cx\sqrt{x} + dx^2 \ln x + \frac{e}{\sqrt{x}}$$

$$4199 \ y = a + bx + cx\sqrt{x} + dx^2 \ln x + e \frac{\ln x}{x}$$

$$4200 \ y = a + bx + cx\sqrt{x} + dx^2 \ln x + \frac{e}{x}$$

$$4201 \ y = a + bx + cx\sqrt{x} + dx^2 \ln x + \frac{e}{x\sqrt{x}}$$

$$4202 \ y = a + bx + cx\sqrt{x} + dx^2 \ln x + e \frac{\ln x}{x^2}$$

$$4203 \ y = a + bx + cx\sqrt{x} + dx^2 \ln x + \frac{e}{x^2}$$

$$4204 \ y = a + bx + cx\sqrt{x} + dx^2 \ln x + e e^{-x}$$

$$4205 \ y = a + bx + cx\sqrt{x} + dx^2 \sqrt{x} + ex^3$$

$$4206 \ y = a + bx + cx\sqrt{x} + dx^2 \sqrt{x} + e e^x$$

$$4207 \ y = a + bx + cx\sqrt{x} + dx^2 \sqrt{x} + e\sqrt{x} \ln x$$

$$4208 \ y = a + bx + cx\sqrt{x} + dx^2 \sqrt{x} + e(\ln x)^2$$

$$4209 \ y = a + bx + cx\sqrt{x} + dx^2 \sqrt{x} + e \frac{x}{\ln x}$$

$$4210 \ y = a + bx + cx\sqrt{x} + dx^2 \sqrt{x} + e\sqrt{x}$$

$$4211 \ y = a + bx + cx\sqrt{x} + dx^2 \sqrt{x} + e \ln x$$

$$4212 \ y = a + bx + cx\sqrt{x} + dx^2 \sqrt{x} + \frac{e}{\ln x}$$

$$4213 \ y = a + bx + cx\sqrt{x} + dx^2 \sqrt{x} + \frac{e}{\sqrt{x}}$$

$$4214 \ y = a + bx + cx\sqrt{x} + dx^2 \sqrt{x} + e \frac{\ln x}{x}$$

$$4215 \ y = a + bx + cx\sqrt{x} + dx^2 \sqrt{x} + \frac{e}{x}$$

$$4216 \ y = a + bx + cx\sqrt{x} + dx^2 \sqrt{x} + \frac{e}{x\sqrt{x}}$$

$$4217 \ y = a + bx + cx\sqrt{x} + dx^2 \sqrt{x} + e \frac{\ln x}{x^2}$$

$$4218 \ y = a + bx + cx\sqrt{x} + dx^2 \sqrt{x} + \frac{e}{x^2}$$

$$4219 \ y = a + bx + cx\sqrt{x} + dx^2 \sqrt{x} + e e^{-x}$$

$$4220 \ y = a + bx + cx\sqrt{x} + dx^3 + e e^x$$

$$4221 \ y = a + bx + cx\sqrt{x} + dx^3 + e\sqrt{x} \ln x$$

$$4222 \ y = a + bx + cx\sqrt{x} + dx^3 + e(\ln x)^2$$

$$4223 \ y = a + bx + cx\sqrt{x} + dx^3 + e \frac{x}{\ln x}$$

$$4224 \ y = a + bx + cx\sqrt{x} + dx^3 + e\sqrt{x}$$

$$4225 \ y = a + bx + cx\sqrt{x} + dx^3 + e \ln x$$

$$4226 \ y = a + bx + cx\sqrt{x} + dx^3 + \frac{e}{\ln x}$$

$$4227 \ y = a + bx + cx\sqrt{x} + dx^3 + \frac{e}{\sqrt{x}}$$

$$4228 \ y = a + bx + cx\sqrt{x} + dx^3 + e \frac{\ln x}{x}$$

- 4229 $y = a + bx + cx\sqrt{x} + dx^3 + \frac{e}{x}$
- 4230 $y = a + bx + cx\sqrt{x} + dx^3 + \frac{e}{x\sqrt{x}}$
- 4231 $y = a + bx + cx\sqrt{x} + dx^3 + e\frac{\ln x}{x^2}$
- 4232 $y = a + bx + cx\sqrt{x} + dx^3 + \frac{e}{x^2}$
- 4233 $y = a + bx + cx\sqrt{x} + dx^3 + e e^{-x}$
- 4234 $y = a + bx + cx\sqrt{x} + d e^x + e\sqrt{x} \ln x$
- 4235 $y = a + bx + cx\sqrt{x} + d e^x + e(\ln x)^2$
- 4236 $y = a + bx + cx\sqrt{x} + d e^x + e\frac{x}{\ln x}$
- 4237 $y = a + bx + cx\sqrt{x} + d e^x + e\sqrt{x}$
- 4238 $y = a + bx + cx\sqrt{x} + d e^x + e \ln x$
- 4239 $y = a + bx + cx\sqrt{x} + d e^x + \frac{e}{\ln x}$
- 4240 $y = a + bx + cx\sqrt{x} + d e^x + \frac{e}{\sqrt{x}}$
- 4241 $y = a + bx + cx\sqrt{x} + d e^x + e\frac{\ln x}{x}$
- 4242 $y = a + bx + cx\sqrt{x} + d e^x + \frac{e}{x}$
- 4243 $y = a + bx + cx\sqrt{x} + d e^x + \frac{e}{x\sqrt{x}}$
- 4244 $y = a + bx + cx\sqrt{x} + d e^x + e\frac{\ln x}{x^2}$
- 4245 $y = a + bx + cx\sqrt{x} + d e^x + \frac{e}{x^2}$
- 4246 $y = a + bx + cx\sqrt{x} + d e^x + e e^{-x}$
- 4247 $y = a + bx + cx\sqrt{x} + d\sqrt{x} \ln x + e(\ln x)^2$
- 4248 $y = a + bx + cx\sqrt{x} + d\sqrt{x} \ln x + e\frac{x}{\ln x}$
- 4249 $y = a + bx + cx\sqrt{x} + d\sqrt{x} \ln x + e\sqrt{x}$
- 4250 $y = a + bx + cx\sqrt{x} + d\sqrt{x} \ln x + e \ln x$
- 4251 $y = a + bx + cx\sqrt{x} + d\sqrt{x} \ln x + \frac{e}{\ln x}$
- 4252 $y = a + bx + cx\sqrt{x} + d\sqrt{x} \ln x + \frac{e}{\sqrt{x}}$
- 4253 $y = a + bx + cx\sqrt{x} + d\sqrt{x} \ln x + e\frac{\ln x}{x}$
- 4254 $y = a + bx + cx\sqrt{x} + d\sqrt{x} \ln x + \frac{e}{x}$
- 4255 $y = a + bx + cx\sqrt{x} + d\sqrt{x} \ln x + \frac{e}{x\sqrt{x}}$
- 4256 $y = a + bx + cx\sqrt{x} + d\sqrt{x} \ln x + e\frac{\ln x}{x^2}$
- 4257 $y = a + bx + cx\sqrt{x} + d\sqrt{x} \ln x + \frac{e}{x^2}$
- 4258 $y = a + bx + cx\sqrt{x} + d\sqrt{x} \ln x + e e^{-x}$
- 4259 $y = a + bx + cx\sqrt{x} + d(\ln x)^2 + e\frac{x}{\ln x}$
- 4260 $y = a + bx + cx\sqrt{x} + d(\ln x)^2 + e\sqrt{x}$
- 4261 $y = a + bx + cx\sqrt{x} + d(\ln x)^2 + e \ln x$
- 4262 $y = a + bx + cx\sqrt{x} + d(\ln x)^2 + \frac{e}{\ln x}$
- 4263 $y = a + bx + cx\sqrt{x} + d(\ln x)^2 + \frac{e}{\sqrt{x}}$
- 4264 $y = a + bx + cx\sqrt{x} + d(\ln x)^2 + e\frac{\ln x}{x}$
- 4265 $y = a + bx + cx\sqrt{x} + d(\ln x)^2 + \frac{e}{x}$
- 4266 $y = a + bx + cx\sqrt{x} + d(\ln x)^2 + \frac{e}{x\sqrt{x}}$
- 4267 $y = a + bx + cx\sqrt{x} + d(\ln x)^2 + e\frac{\ln x}{x^2}$
- 4268 $y = a + bx + cx\sqrt{x} + d(\ln x)^2 + \frac{e}{x^2}$
- 4269 $y = a + bx + cx\sqrt{x} + d(\ln x)^2 + e e^{-x}$
- 4270 $y = a + bx + cx\sqrt{x} + d\frac{x}{\ln x} + e\sqrt{x}$
- 4271 $y = a + bx + cx\sqrt{x} + d\frac{x}{\ln x} + e \ln x$
- 4272 $y = a + bx + cx\sqrt{x} + d\frac{x}{\ln x} + \frac{e}{\ln x}$
- 4273 $y = a + bx + cx\sqrt{x} + d\frac{x}{\ln x} + \frac{e}{\sqrt{x}}$
- 4274 $y = a + bx + cx\sqrt{x} + d\frac{x}{\ln x} + e\frac{\ln x}{x}$
- 4275 $y = a + bx + cx\sqrt{x} + d\frac{x}{\ln x} + \frac{e}{x}$
- 4276 $y = a + bx + cx\sqrt{x} + d\frac{x}{\ln x} + \frac{e}{x\sqrt{x}}$
- 4277 $y = a + bx + cx\sqrt{x} + d\frac{x}{\ln x} + e\frac{\ln x}{x^2}$
- 4278 $y = a + bx + cx\sqrt{x} + d\frac{x}{\ln x} + \frac{e}{x^2}$
- 4279 $y = a + bx + cx\sqrt{x} + d\frac{x}{\ln x} + e e^{-x}$
- 4280 $y = a + bx + cx\sqrt{x} + d\sqrt{x} + e \ln x$
- 4281 $y = a + bx + cx\sqrt{x} + d\sqrt{x} + \frac{e}{\ln x}$
- 4282 $y = a + bx + cx\sqrt{x} + d\sqrt{x} + \frac{e}{\sqrt{x}}$
- 4283 $y = a + bx + cx\sqrt{x} + d\sqrt{x} + e\frac{\ln x}{x}$
- 4284 $y = a + bx + cx\sqrt{x} + d\sqrt{x} + \frac{e}{x}$
- 4285 $y = a + bx + cx\sqrt{x} + d\sqrt{x} + \frac{e}{x\sqrt{x}}$
- 4286 $y = a + bx + cx\sqrt{x} + d\sqrt{x} + e\frac{\ln x}{x^2}$
- 4287 $y = a + bx + cx\sqrt{x} + d\sqrt{x} + \frac{e}{x^2}$
- 4288 $y = a + bx + cx\sqrt{x} + d\sqrt{x} + e e^{-x}$
- 4289 $y = a + bx + cx\sqrt{x} + d \ln x + \frac{e}{\ln x}$
- 4290 $y = a + bx + cx\sqrt{x} + d \ln x + \frac{e}{\sqrt{x}}$
- 4291 $y = a + bx + cx\sqrt{x} + d \ln x + e\frac{\ln x}{x}$
- 4292 $y = a + bx + cx\sqrt{x} + d \ln x + \frac{e}{x}$
- 4293 $y = a + bx + cx\sqrt{x} + d \ln x + \frac{e}{x\sqrt{x}}$
- 4294 $y = a + bx + cx\sqrt{x} + d \ln x + e\frac{\ln x}{x^2}$
- 4295 $y = a + bx + cx\sqrt{x} + d \ln x + \frac{e}{x^2}$
- 4296 $y = a + bx + cx\sqrt{x} + d \ln x + e e^{-x}$
- 4297 $y = a + bx + cx\sqrt{x} + \frac{d}{\ln x} + \frac{e}{\sqrt{x}}$
- 4298 $y = a + bx + cx\sqrt{x} + \frac{d}{\ln x} + e\frac{\ln x}{x}$
- 4299 $y = a + bx + cx\sqrt{x} + \frac{d}{\ln x} + \frac{e}{x}$
- 4300 $y = a + bx + cx\sqrt{x} + \frac{d}{\ln x} + \frac{e}{x\sqrt{x}}$
- 4301 $y = a + bx + cx\sqrt{x} + \frac{d}{\ln x} + e\frac{\ln x}{x^2}$
- 4302 $y = a + bx + cx\sqrt{x} + \frac{d}{\ln x} + \frac{e}{x^2}$
- 4303 $y = a + bx + cx\sqrt{x} + \frac{d}{\ln x} + e e^{-x}$
- 4304 $y = a + bx + cx\sqrt{x} + \frac{d}{\sqrt{x}} + e\frac{\ln x}{x}$
- 4305 $y = a + bx + cx\sqrt{x} + \frac{d}{\sqrt{x}} + \frac{e}{x}$
- 4306 $y = a + bx + cx\sqrt{x} + \frac{d}{\sqrt{x}} + \frac{e}{x\sqrt{x}}$

- 4307 $y = a + bx + cx\sqrt{x} + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x^2}$
- 4308 $y = a + bx + cx\sqrt{x} + \frac{d}{\sqrt{x}} + \frac{e}{x^2}$
- 4309 $y = a + bx + cx\sqrt{x} + \frac{d}{\sqrt{x}} + e e^{-x}$
- 4310 $y = a + bx + cx\sqrt{x} + d \frac{\ln x}{x} + \frac{e}{x}$
- 4311 $y = a + bx + cx\sqrt{x} + d \frac{\ln x}{x} + \frac{e}{x\sqrt{x}}$
- 4312 $y = a + bx + cx\sqrt{x} + d \frac{\ln x}{x} + e \frac{\ln x}{x^2}$
- 4313 $y = a + bx + cx\sqrt{x} + d \frac{\ln x}{x} + \frac{e}{x^2}$
- 4314 $y = a + bx + cx\sqrt{x} + d \frac{\ln x}{x} + e e^{-x}$
- 4315 $y = a + bx + cx\sqrt{x} + \frac{d}{x} + \frac{e}{x\sqrt{x}}$
- 4316 $y = a + bx + cx\sqrt{x} + \frac{d}{x} + e \frac{\ln x}{x^2}$
- 4317 $y = a + bx + cx\sqrt{x} + \frac{d}{x} + \frac{e}{x^2}$
- 4318 $y = a + bx + cx\sqrt{x} + \frac{d}{x} + e e^{-x}$
- 4319 $y = a + bx + cx\sqrt{x} + \frac{d}{x\sqrt{x}} + e \frac{\ln x}{x^2}$
- 4320 $y = a + bx + cx\sqrt{x} + \frac{d}{x\sqrt{x}} + \frac{e}{x^2}$
- 4321 $y = a + bx + cx\sqrt{x} + \frac{d}{x\sqrt{x}} + e e^{-x}$
- 4322 $y = a + bx + cx\sqrt{x} + d \frac{\ln x}{x^2} + \frac{e}{x^2}$
- 4323 $y = a + bx + cx\sqrt{x} + d \frac{\ln x}{x^2} + e e^{-x}$
- 4324 $y = a + bx + cx\sqrt{x} + \frac{d}{x^2} + e e^{-x}$
- 4325 $y = a + bx + cx^2 + dx^2 \ln x + ex^2 \sqrt{x}$
- 4326 $y = a + bx + cx^2 + dx^2 \ln x + ex^3$
- 4327 $y = a + bx + cx^2 + dx^2 \ln x + e e^x$
- 4328 $y = a + bx + cx^2 + dx^2 \ln x + e\sqrt{x} \ln x$
- 4329 $y = a + bx + cx^2 + dx^2 \ln x + e(\ln x)^2$
- 4330 $y = a + bx + cx^2 + dx^2 \ln x + e \frac{x}{\ln x}$
- 4331 $y = a + bx + cx^2 + dx^2 \ln x + e\sqrt{x}$
- 4332 $y = a + bx + cx^2 + dx^2 \ln x + e \ln x$
- 4333 $y = a + bx + cx^2 + dx^2 \ln x + \frac{e}{\ln x}$
- 4334 $y = a + bx + cx^2 + dx^2 \ln x + \frac{e}{\sqrt{x}}$
- 4335 $y = a + bx + cx^2 + dx^2 \ln x + e \frac{\ln x}{x}$
- 4336 $y = a + bx + cx^2 + dx^2 \ln x + \frac{e}{x}$
- 4337 $y = a + bx + cx^2 + dx^2 \ln x + \frac{e}{x\sqrt{x}}$
- 4338 $y = a + bx + cx^2 + dx^2 \ln x + e \frac{\ln x}{x^2}$
- 4339 $y = a + bx + cx^2 + dx^2 \ln x + \frac{e}{x^2}$
- 4340 $y = a + bx + cx^2 + dx^2 \ln x + e e^{-x}$
- 4341 $y = a + bx + cx^2 + dx^2 \sqrt{x} + ex^3$
- 4342 $y = a + bx + cx^2 + dx^2 \sqrt{x} + e e^x$
- 4343 $y = a + bx + cx^2 + dx^2 \sqrt{x} + e\sqrt{x} \ln x$
- 4344 $y = a + bx + cx^2 + dx^2 \sqrt{x} + e(\ln x)^2$
- 4345 $y = a + bx + cx^2 + dx^2 \sqrt{x} + e \frac{x}{\ln x}$
- 4346 $y = a + bx + cx^2 + dx^2 \sqrt{x} + e\sqrt{x}$
- 4347 $y = a + bx + cx^2 + dx^2 \sqrt{x} + e \ln x$
- 4348 $y = a + bx + cx^2 + dx^2 \sqrt{x} + \frac{e}{\ln x}$
- 4349 $y = a + bx + cx^2 + dx^2 \sqrt{x} + \frac{e}{\sqrt{x}}$
- 4350 $y = a + bx + cx^2 + dx^2 \sqrt{x} + e \frac{\ln x}{x}$
- 4351 $y = a + bx + cx^2 + dx^2 \sqrt{x} + \frac{e}{x}$
- 4352 $y = a + bx + cx^2 + dx^2 \sqrt{x} + \frac{e}{x\sqrt{x}}$
- 4353 $y = a + bx + cx^2 + dx^2 \sqrt{x} + e \frac{\ln x}{x^2}$
- 4354 $y = a + bx + cx^2 + dx^2 \sqrt{x} + \frac{e}{x^2}$
- 4355 $y = a + bx + cx^2 + dx^2 \sqrt{x} + e e^{-x}$
- 4356 $y = a + bx + cx^2 + dx^3 + e e^x$
- 4357 $y = a + bx + cx^2 + dx^3 + e\sqrt{x} \ln x$
- 4358 $y = a + bx + cx^2 + dx^3 + e(\ln x)^2$
- 4359 $y = a + bx + cx^2 + dx^3 + e \frac{x}{\ln x}$
- 4360 $y = a + bx + cx^2 + dx^3 + e\sqrt{x}$
- 4361 $y = a + bx + cx^2 + dx^3 + e \ln x$
- 4362 $y = a + bx + cx^2 + dx^3 + \frac{e}{\ln x}$
- 4363 $y = a + bx + cx^2 + dx^3 + \frac{e}{\sqrt{x}}$
- 4364 $y = a + bx + cx^2 + dx^3 + e \frac{\ln x}{x}$
- 4365 $y = a + bx + cx^2 + dx^3 + \frac{e}{x}$
- 4366 $y = a + bx + cx^2 + dx^3 + \frac{e}{x\sqrt{x}}$
- 4367 $y = a + bx + cx^2 + dx^3 + e \frac{\ln x}{x^2}$
- 4368 $y = a + bx + cx^2 + dx^3 + \frac{e}{x^2}$
- 4369 $y = a + bx + cx^2 + dx^3 + e e^{-x}$
- 4370 $y = a + bx + cx^2 + d e^x + e\sqrt{x} \ln x$
- 4371 $y = a + bx + cx^2 + d e^x + e(\ln x)^2$
- 4372 $y = a + bx + cx^2 + d e^x + e \frac{x}{\ln x}$
- 4373 $y = a + bx + cx^2 + d e^x + e\sqrt{x}$
- 4374 $y = a + bx + cx^2 + d e^x + e \ln x$
- 4375 $y = a + bx + cx^2 + d e^x + \frac{e}{\ln x}$
- 4376 $y = a + bx + cx^2 + d e^x + \frac{e}{\sqrt{x}}$
- 4377 $y = a + bx + cx^2 + d e^x + e \frac{\ln x}{x}$
- 4378 $y = a + bx + cx^2 + d e^x + \frac{e}{x}$
- 4379 $y = a + bx + cx^2 + d e^x + \frac{e}{x\sqrt{x}}$
- 4380 $y = a + bx + cx^2 + d e^x + e \frac{\ln x}{x^2}$
- 4381 $y = a + bx + cx^2 + d e^x + \frac{e}{x^2}$
- 4382 $y = a + bx + cx^2 + d e^x + e e^{-x}$
- 4383 $y = a + bx + cx^2 + d\sqrt{x} \ln x + e(\ln x)^2$
- 4384 $y = a + bx + cx^2 + d\sqrt{x} \ln x + e \frac{x}{\ln x}$

- 4385 $y = a + bx + cx^2 + d\sqrt{x} \ln x + e\sqrt{x}$
- 4386 $y = a + bx + cx^2 + d\sqrt{x} \ln x + e \ln x$
- 4387 $y = a + bx + cx^2 + d\sqrt{x} \ln x + \frac{e}{\ln x}$
- 4388 $y = a + bx + cx^2 + d\sqrt{x} \ln x + \frac{e}{\sqrt{x}}$
- 4389 $y = a + bx + cx^2 + d\sqrt{x} \ln x + e \frac{\ln x}{x}$
- 4390 $y = a + bx + cx^2 + d\sqrt{x} \ln x + \frac{e}{x}$
- 4391 $y = a + bx + cx^2 + d\sqrt{x} \ln x + \frac{e}{x\sqrt{x}}$
- 4392 $y = a + bx + cx^2 + d\sqrt{x} \ln x + e \frac{\ln x}{x^2}$
- 4393 $y = a + bx + cx^2 + d\sqrt{x} \ln x + \frac{e}{x^2}$
- 4394 $y = a + bx + cx^2 + d\sqrt{x} \ln x + e e^{-x}$
- 4395 $y = a + bx + cx^2 + d(\ln x)^2 + e \frac{x}{\ln x}$
- 4396 $y = a + bx + cx^2 + d(\ln x)^2 + e\sqrt{x}$
- 4397 $y = a + bx + cx^2 + d(\ln x)^2 + e \ln x$
- 4398 $y = a + bx + cx^2 + d(\ln x)^2 + \frac{e}{\ln x}$
- 4399 $y = a + bx + cx^2 + d(\ln x)^2 + \frac{e}{\sqrt{x}}$
- 4400 $y = a + bx + cx^2 + d(\ln x)^2 + e \frac{\ln x}{x}$
- 4401 $y = a + bx + cx^2 + d(\ln x)^2 + \frac{e}{x}$
- 4402 $y = a + bx + cx^2 + d(\ln x)^2 + \frac{e}{x\sqrt{x}}$
- 4403 $y = a + bx + cx^2 + d(\ln x)^2 + e \frac{\ln x}{x^2}$
- 4404 $y = a + bx + cx^2 + d(\ln x)^2 + \frac{e}{x^2}$
- 4405 $y = a + bx + cx^2 + d(\ln x)^2 + e e^{-x}$
- 4406 $y = a + bx + cx^2 + d \frac{x}{\ln x} + e\sqrt{x}$
- 4407 $y = a + bx + cx^2 + d \frac{x}{\ln x} + e \ln x$
- 4408 $y = a + bx + cx^2 + d \frac{x}{\ln x} + \frac{e}{\ln x}$
- 4409 $y = a + bx + cx^2 + d \frac{x}{\ln x} + \frac{e}{\sqrt{x}}$
- 4410 $y = a + bx + cx^2 + d \frac{x}{\ln x} + e \frac{\ln x}{x}$
- 4411 $y = a + bx + cx^2 + d \frac{x}{\ln x} + \frac{e}{x}$
- 4412 $y = a + bx + cx^2 + d \frac{x}{\ln x} + \frac{e}{x\sqrt{x}}$
- 4413 $y = a + bx + cx^2 + d \frac{x}{\ln x} + e \frac{\ln x}{x^2}$
- 4414 $y = a + bx + cx^2 + d \frac{x}{\ln x} + \frac{e}{x^2}$
- 4415 $y = a + bx + cx^2 + d \frac{x}{\ln x} + e e^{-x}$
- 4416 $y = a + bx + cx^2 + d\sqrt{x} + e \ln x$
- 4417 $y = a + bx + cx^2 + d\sqrt{x} + \frac{e}{\ln x}$
- 4418 $y = a + bx + cx^2 + d\sqrt{x} + \frac{e}{\sqrt{x}}$
- 4419 $y = a + bx + cx^2 + d\sqrt{x} + e \frac{\ln x}{x}$
- 4420 $y = a + bx + cx^2 + d\sqrt{x} + \frac{e}{x}$
- 4421 $y = a + bx + cx^2 + d\sqrt{x} + \frac{e}{x\sqrt{x}}$
- 4422 $y = a + bx + cx^2 + d\sqrt{x} + e \frac{\ln x}{x^2}$
- 4423 $y = a + bx + cx^2 + d\sqrt{x} + \frac{e}{x^2}$
- 4424 $y = a + bx + cx^2 + d\sqrt{x} + e e^{-x}$
- 4425 $y = a + bx + cx^2 + d \ln x + \frac{e}{\ln x}$
- 4426 $y = a + bx + cx^2 + d \ln x + \frac{e}{\sqrt{x}}$
- 4427 $y = a + bx + cx^2 + d \ln x + e \frac{\ln x}{x}$
- 4428 $y = a + bx + cx^2 + d \ln x + \frac{e}{x}$
- 4429 $y = a + bx + cx^2 + d \ln x + \frac{e}{x\sqrt{x}}$
- 4430 $y = a + bx + cx^2 + d \ln x + e \frac{\ln x}{x^2}$
- 4431 $y = a + bx + cx^2 + d \ln x + \frac{e}{x^2}$
- 4432 $y = a + bx + cx^2 + d \ln x + e e^{-x}$
- 4433 $y = a + bx + cx^2 + \frac{d}{\ln x} + \frac{e}{\sqrt{x}}$
- 4434 $y = a + bx + cx^2 + \frac{d}{\ln x} + e \frac{\ln x}{x}$
- 4435 $y = a + bx + cx^2 + \frac{d}{\ln x} + \frac{e}{x}$
- 4436 $y = a + bx + cx^2 + \frac{d}{\ln x} + \frac{e}{x\sqrt{x}}$
- 4437 $y = a + bx + cx^2 + \frac{d}{\ln x} + e \frac{\ln x}{x^2}$
- 4438 $y = a + bx + cx^2 + \frac{d}{\ln x} + \frac{e}{x^2}$
- 4439 $y = a + bx + cx^2 + \frac{d}{\ln x} + e e^{-x}$
- 4440 $y = a + bx + cx^2 + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x}$
- 4441 $y = a + bx + cx^2 + \frac{d}{\sqrt{x}} + \frac{e}{x}$
- 4442 $y = a + bx + cx^2 + \frac{d}{\sqrt{x}} + \frac{e}{x\sqrt{x}}$
- 4443 $y = a + bx + cx^2 + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x^2}$
- 4444 $y = a + bx + cx^2 + \frac{d}{\sqrt{x}} + \frac{e}{x^2}$
- 4445 $y = a + bx + cx^2 + \frac{d}{\sqrt{x}} + e e^{-x}$
- 4446 $y = a + bx + cx^2 + d \frac{\ln x}{x} + \frac{e}{x}$
- 4447 $y = a + bx + cx^2 + d \frac{\ln x}{x} + \frac{e}{x\sqrt{x}}$
- 4448 $y = a + bx + cx^2 + d \frac{\ln x}{x} + e \frac{\ln x}{x^2}$
- 4449 $y = a + bx + cx^2 + d \frac{\ln x}{x} + \frac{e}{x^2}$
- 4450 $y = a + bx + cx^2 + d \frac{\ln x}{x} + e e^{-x}$
- 4451 $y = a + bx + cx^2 + \frac{d}{x} + \frac{e}{x\sqrt{x}}$
- 4452 $y = a + bx + cx^2 + \frac{d}{x} + e \frac{\ln x}{x^2}$
- 4453 $y = a + bx + cx^2 + \frac{d}{x} + \frac{e}{x^2}$
- 4454 $y = a + bx + cx^2 + \frac{d}{x} + e e^{-x}$
- 4455 $y = a + bx + cx^2 + \frac{d}{x\sqrt{x}} + e \frac{\ln x}{x^2}$
- 4456 $y = a + bx + cx^2 + \frac{d}{x\sqrt{x}} + \frac{e}{x^2}$
- 4457 $y = a + bx + cx^2 + \frac{d}{x\sqrt{x}} + e e^{-x}$
- 4458 $y = a + bx + cx^2 + d \frac{\ln x}{x^2} + \frac{e}{x^2}$
- 4459 $y = a + bx + cx^2 + d \frac{\ln x}{x^2} + e e^{-x}$
- 4460 $y = a + bx + cx^2 + \frac{d}{x^2} + e e^{-x}$
- 4461 $y = a + bx + cx^2 \ln x + dx^2 \sqrt{x} + ex^3$
- 4462 $y = a + bx + cx^2 \ln x + dx^2 \sqrt{x} + e e^x$

- 4463 $y = a + bx + cx^2 \ln x + dx^2 \sqrt{x} + e\sqrt{x} \ln x$ 4489 $y = a + bx + cx^2 \ln x + dx^3 + e e^{-x}$ 4515 $y = a + bx + cx^2 \ln x + d(\ln x)^2 + e \frac{x}{\ln x}$
- 4464 $y = a + bx + cx^2 \ln x + dx^2 \sqrt{x} + e(\ln x)^2$ 4490 $y = a + bx + cx^2 \ln x + d e^x + e\sqrt{x} \ln x$ 4516 $y = a + bx + cx^2 \ln x + d(\ln x)^2 + e\sqrt{x}$
- 4465 $y = a + bx + cx^2 \ln x + dx^2 \sqrt{x} + e \frac{x}{\ln x}$ 4491 $y = a + bx + cx^2 \ln x + d e^x + e(\ln x)^2$ 4517 $y = a + bx + cx^2 \ln x + d(\ln x)^2 + e \ln x$
- 4466 $y = a + bx + cx^2 \ln x + dx^2 \sqrt{x} + e\sqrt{x}$ 4492 $y = a + bx + cx^2 \ln x + d e^x + e \frac{x}{\ln x}$ 4518 $y = a + bx + cx^2 \ln x + d(\ln x)^2 + \frac{e}{\ln x}$
- 4467 $y = a + bx + cx^2 \ln x + dx^2 \sqrt{x} + e \ln x$ 4493 $y = a + bx + cx^2 \ln x + d e^x + e\sqrt{x}$ 4519 $y = a + bx + cx^2 \ln x + d(\ln x)^2 + \frac{e}{\sqrt{x}}$
- 4468 $y = a + bx + cx^2 \ln x + dx^2 \sqrt{x} + \frac{e}{\ln x}$ 4494 $y = a + bx + cx^2 \ln x + d e^x + e \ln x$ 4520 $y = a + bx + cx^2 \ln x + d(\ln x)^2 + e \frac{\ln x}{x}$
- 4469 $y = a + bx + cx^2 \ln x + dx^2 \sqrt{x} + \frac{e}{\sqrt{x}}$ 4495 $y = a + bx + cx^2 \ln x + d e^x + \frac{e}{\ln x}$ 4521 $y = a + bx + cx^2 \ln x + d(\ln x)^2 + \frac{e}{x}$
- 4470 $y = a + bx + cx^2 \ln x + dx^2 \sqrt{x} + e \frac{\ln x}{x}$ 4496 $y = a + bx + cx^2 \ln x + d e^x + \frac{e}{\sqrt{x}}$ 4522 $y = a + bx + cx^2 \ln x + d(\ln x)^2 + \frac{e}{x\sqrt{x}}$
- 4471 $y = a + bx + cx^2 \ln x + dx^2 \sqrt{x} + \frac{e}{x}$ 4497 $y = a + bx + cx^2 \ln x + d e^x + e \frac{\ln x}{x}$ 4523 $y = a + bx + cx^2 \ln x + d(\ln x)^2 + e \frac{\ln x}{x^2}$
- 4472 $y = a + bx + cx^2 \ln x + dx^2 \sqrt{x} + \frac{e}{x\sqrt{x}}$ 4498 $y = a + bx + cx^2 \ln x + d e^x + \frac{e}{x}$ 4524 $y = a + bx + cx^2 \ln x + d(\ln x)^2 + \frac{e}{x^2}$
- 4473 $y = a + bx + cx^2 \ln x + dx^2 \sqrt{x} + e \frac{\ln x}{x^2}$ 4499 $y = a + bx + cx^2 \ln x + d e^x + \frac{e}{x\sqrt{x}}$ 4525 $y = a + bx + cx^2 \ln x + d(\ln x)^2 + e e^{-x}$
- 4474 $y = a + bx + cx^2 \ln x + dx^2 \sqrt{x} + \frac{e}{x^2}$ 4500 $y = a + bx + cx^2 \ln x + d e^x + e \frac{\ln x}{x^2}$ 4526 $y = a + bx + cx^2 \ln x + d \frac{x}{\ln x} + e\sqrt{x}$
- 4475 $y = a + bx + cx^2 \ln x + dx^2 \sqrt{x} + e e^{-x}$ 4501 $y = a + bx + cx^2 \ln x + d e^x + \frac{e}{x^2}$ 4527 $y = a + bx + cx^2 \ln x + d \frac{x}{\ln x} + e \ln x$
- 4476 $y = a + bx + cx^2 \ln x + dx^3 + e e^x$ 4502 $y = a + bx + cx^2 \ln x + d e^x + e e^{-x}$ 4528 $y = a + bx + cx^2 \ln x + d \frac{x}{\ln x} + \frac{e}{\ln x}$
- 4477 $y = a + bx + cx^2 \ln x + dx^3 + e\sqrt{x} \ln x$ 4503 $y = a + bx + cx^2 \ln x + d\sqrt{x} \ln x + e(\ln x)^2$ 4529 $y = a + bx + cx^2 \ln x + d \frac{x}{\ln x} + \frac{e}{\sqrt{x}}$
- 4478 $y = a + bx + cx^2 \ln x + dx^3 + e(\ln x)^2$ 4504 $y = a + bx + cx^2 \ln x + d\sqrt{x} \ln x + e \frac{x}{\ln x}$ 4530 $y = a + bx + cx^2 \ln x + d \frac{x}{\ln x} + e \frac{\ln x}{x}$
- 4479 $y = a + bx + cx^2 \ln x + dx^3 + e \frac{x}{\ln x}$ 4505 $y = a + bx + cx^2 \ln x + d\sqrt{x} \ln x + e\sqrt{x}$ 4531 $y = a + bx + cx^2 \ln x + d \frac{x}{\ln x} + \frac{e}{x}$
- 4480 $y = a + bx + cx^2 \ln x + dx^3 + e\sqrt{x}$ 4506 $y = a + bx + cx^2 \ln x + d\sqrt{x} \ln x + e \ln x$ 4532 $y = a + bx + cx^2 \ln x + d \frac{x}{\ln x} + \frac{e}{x\sqrt{x}}$
- 4481 $y = a + bx + cx^2 \ln x + dx^3 + e \ln x$ 4507 $y = a + bx + cx^2 \ln x + d\sqrt{x} \ln x + \frac{e}{\ln x}$ 4533 $y = a + bx + cx^2 \ln x + d \frac{x}{\ln x} + e \frac{\ln x}{x^2}$
- 4482 $y = a + bx + cx^2 \ln x + dx^3 + \frac{e}{\ln x}$ 4508 $y = a + bx + cx^2 \ln x + d\sqrt{x} \ln x + \frac{e}{\sqrt{x}}$ 4534 $y = a + bx + cx^2 \ln x + d \frac{x}{\ln x} + \frac{e}{x^2}$
- 4483 $y = a + bx + cx^2 \ln x + dx^3 + \frac{e}{\sqrt{x}}$ 4509 $y = a + bx + cx^2 \ln x + d\sqrt{x} \ln x + e \frac{\ln x}{x}$ 4535 $y = a + bx + cx^2 \ln x + d \frac{x}{\ln x} + e e^{-x}$
- 4484 $y = a + bx + cx^2 \ln x + dx^3 + e \frac{\ln x}{x}$ 4510 $y = a + bx + cx^2 \ln x + d\sqrt{x} \ln x + \frac{e}{x}$ 4536 $y = a + bx + cx^2 \ln x + d\sqrt{x} + e \ln x$
- 4485 $y = a + bx + cx^2 \ln x + dx^3 + \frac{e}{x}$ 4511 $y = a + bx + cx^2 \ln x + d\sqrt{x} \ln x + \frac{e}{x\sqrt{x}}$ 4537 $y = a + bx + cx^2 \ln x + d\sqrt{x} + \frac{e}{\ln x}$
- 4486 $y = a + bx + cx^2 \ln x + dx^3 + \frac{e}{x\sqrt{x}}$ 4512 $y = a + bx + cx^2 \ln x + d\sqrt{x} \ln x + e \frac{\ln x}{x^2}$ 4538 $y = a + bx + cx^2 \ln x + d\sqrt{x} + \frac{e}{\sqrt{x}}$
- 4487 $y = a + bx + cx^2 \ln x + dx^3 + e \frac{\ln x}{x^2}$ 4513 $y = a + bx + cx^2 \ln x + d\sqrt{x} \ln x + \frac{e}{x^2}$ 4539 $y = a + bx + cx^2 \ln x + d\sqrt{x} + e \frac{\ln x}{x}$
- 4488 $y = a + bx + cx^2 \ln x + dx^3 + \frac{e}{x^2}$ 4514 $y = a + bx + cx^2 \ln x + d\sqrt{x} \ln x + e e^{-x}$ 4540 $y = a + bx + cx^2 \ln x + d\sqrt{x} + \frac{e}{x}$

- 4541 $y = a + bx + cx^2 \ln x + d\sqrt{x} + \frac{e}{x\sqrt{x}}$
- 4542 $y = a + bx + cx^2 \ln x + d\sqrt{x} + e\frac{\ln x}{x^2}$
- 4543 $y = a + bx + cx^2 \ln x + d\sqrt{x} + \frac{e}{x^2}$
- 4544 $y = a + bx + cx^2 \ln x + d\sqrt{x} + e e^{-x}$
- 4545 $y = a + bx + cx^2 \ln x + d \ln x + \frac{e}{\ln x}$
- 4546 $y = a + bx + cx^2 \ln x + d \ln x + \frac{e}{\sqrt{x}}$
- 4547 $y = a + bx + cx^2 \ln x + d \ln x + e\frac{\ln x}{x}$
- 4548 $y = a + bx + cx^2 \ln x + d \ln x + \frac{e}{x}$
- 4549 $y = a + bx + cx^2 \ln x + d \ln x + \frac{e}{x\sqrt{x}}$
- 4550 $y = a + bx + cx^2 \ln x + d \ln x + e\frac{\ln x}{x^2}$
- 4551 $y = a + bx + cx^2 \ln x + d \ln x + \frac{e}{x^2}$
- 4552 $y = a + bx + cx^2 \ln x + d \ln x + e e^{-x}$
- 4553 $y = a + bx + cx^2 \ln x + \frac{d}{\ln x} + \frac{e}{\sqrt{x}}$
- 4554 $y = a + bx + cx^2 \ln x + \frac{d}{\ln x} + e\frac{\ln x}{x}$
- 4555 $y = a + bx + cx^2 \ln x + \frac{d}{\ln x} + \frac{e}{x}$
- 4556 $y = a + bx + cx^2 \ln x + \frac{d}{\ln x} + \frac{e}{x\sqrt{x}}$
- 4557 $y = a + bx + cx^2 \ln x + \frac{d}{\ln x} + e\frac{\ln x}{x^2}$
- 4558 $y = a + bx + cx^2 \ln x + \frac{d}{\ln x} + \frac{e}{x^2}$
- 4559 $y = a + bx + cx^2 \ln x + \frac{d}{\ln x} + e e^{-x}$
- 4560 $y = a + bx + cx^2 \ln x + \frac{d}{\sqrt{x}} + e\frac{\ln x}{x}$
- 4561 $y = a + bx + cx^2 \ln x + \frac{d}{\sqrt{x}} + \frac{e}{x}$
- 4562 $y = a + bx + cx^2 \ln x + \frac{d}{\sqrt{x}} + \frac{e}{x\sqrt{x}}$
- 4563 $y = a + bx + cx^2 \ln x + \frac{d}{\sqrt{x}} + e\frac{\ln x}{x^2}$
- 4564 $y = a + bx + cx^2 \ln x + \frac{d}{\sqrt{x}} + \frac{e}{x^2}$
- 4565 $y = a + bx + cx^2 \ln x + \frac{d}{\sqrt{x}} + e e^{-x}$
- 4566 $y = a + bx + cx^2 \ln x + d\frac{\ln x}{x} + \frac{e}{x}$
- 4567 $y = a + bx + cx^2 \ln x + d\frac{\ln x}{x} + \frac{e}{x\sqrt{x}}$
- 4568 $y = a + bx + cx^2 \ln x + d\frac{\ln x}{x} + e\frac{\ln x}{x^2}$
- 4569 $y = a + bx + cx^2 \ln x + d\frac{\ln x}{x} + \frac{e}{x^2}$
- 4570 $y = a + bx + cx^2 \ln x + d\frac{\ln x}{x} + e e^{-x}$
- 4571 $y = a + bx + cx^2 \ln x + \frac{d}{x} + \frac{e}{x\sqrt{x}}$
- 4572 $y = a + bx + cx^2 \ln x + \frac{d}{x} + e\frac{\ln x}{x^2}$
- 4573 $y = a + bx + cx^2 \ln x + \frac{d}{x} + \frac{e}{x^2}$
- 4574 $y = a + bx + cx^2 \ln x + \frac{d}{x} + e e^{-x}$
- 4575 $y = a + bx + cx^2 \ln x + \frac{d}{x\sqrt{x}} + e\frac{\ln x}{x^2}$
- 4576 $y = a + bx + cx^2 \ln x + \frac{d}{x\sqrt{x}} + \frac{e}{x^2}$
- 4577 $y = a + bx + cx^2 \ln x + \frac{d}{x\sqrt{x}} + e e^{-x}$
- 4578 $y = a + bx + cx^2 \ln x + d\frac{\ln x}{x^2} + \frac{e}{x^2}$
- 4579 $y = a + bx + cx^2 \ln x + d\frac{\ln x}{x^2} + e e^{-x}$
- 4580 $y = a + bx + cx^2 \ln x + \frac{d}{x^2} + e e^{-x}$
- 4581 $y = a + bx + cx^2 \sqrt{x} + dx^3 + e e^x$
- 4582 $y = a + bx + cx^2 \sqrt{x} + dx^3 + e\sqrt{x} \ln x$
- 4583 $y = a + bx + cx^2 \sqrt{x} + dx^3 + e(\ln x)^2$
- 4584 $y = a + bx + cx^2 \sqrt{x} + dx^3 + e\frac{x}{\ln x}$
- 4585 $y = a + bx + cx^2 \sqrt{x} + dx^3 + e\sqrt{x}$
- 4586 $y = a + bx + cx^2 \sqrt{x} + dx^3 + e \ln x$
- 4587 $y = a + bx + cx^2 \sqrt{x} + dx^3 + \frac{e}{\ln x}$
- 4588 $y = a + bx + cx^2 \sqrt{x} + dx^3 + \frac{e}{\sqrt{x}}$
- 4589 $y = a + bx + cx^2 \sqrt{x} + dx^3 + e\frac{\ln x}{x}$
- 4590 $y = a + bx + cx^2 \sqrt{x} + dx^3 + \frac{e}{x}$
- 4591 $y = a + bx + cx^2 \sqrt{x} + dx^3 + \frac{e}{x\sqrt{x}}$
- 4592 $y = a + bx + cx^2 \sqrt{x} + dx^3 + e\frac{\ln x}{x^2}$
- 4593 $y = a + bx + cx^2 \sqrt{x} + dx^3 + \frac{e}{x^2}$
- 4594 $y = a + bx + cx^2 \sqrt{x} + dx^3 + e e^{-x}$
- 4595 $y = a + bx + cx^2 \sqrt{x} + d e^x + e\sqrt{x} \ln x$
- 4596 $y = a + bx + cx^2 \sqrt{x} + d e^x + e(\ln x)^2$
- 4597 $y = a + bx + cx^2 \sqrt{x} + d e^x + e\frac{x}{\ln x}$
- 4598 $y = a + bx + cx^2 \sqrt{x} + d e^x + e\sqrt{x}$
- 4599 $y = a + bx + cx^2 \sqrt{x} + d e^x + e \ln x$
- 4600 $y = a + bx + cx^2 \sqrt{x} + d e^x + \frac{e}{\ln x}$
- 4601 $y = a + bx + cx^2 \sqrt{x} + d e^x + \frac{e}{\sqrt{x}}$
- 4602 $y = a + bx + cx^2 \sqrt{x} + d e^x + e\frac{\ln x}{x}$
- 4603 $y = a + bx + cx^2 \sqrt{x} + d e^x + \frac{e}{x}$
- 4604 $y = a + bx + cx^2 \sqrt{x} + d e^x + \frac{e}{x\sqrt{x}}$
- 4605 $y = a + bx + cx^2 \sqrt{x} + d e^x + e\frac{\ln x}{x^2}$
- 4606 $y = a + bx + cx^2 \sqrt{x} + d e^x + \frac{e}{x^2}$
- 4607 $y = a + bx + cx^2 \sqrt{x} + d e^x + e e^{-x}$
- 4608 $y = a + bx + cx^2 \sqrt{x} + d\sqrt{x} \ln x + e(\ln x)^2$
- 4609 $y = a + bx + cx^2 \sqrt{x} + d\sqrt{x} \ln x + e\frac{x}{\ln x}$
- 4610 $y = a + bx + cx^2 \sqrt{x} + d\sqrt{x} \ln x + e\sqrt{x}$
- 4611 $y = a + bx + cx^2 \sqrt{x} + d\sqrt{x} \ln x + e \ln x$
- 4612 $y = a + bx + cx^2 \sqrt{x} + d\sqrt{x} \ln x + \frac{e}{\ln x}$
- 4613 $y = a + bx + cx^2 \sqrt{x} + d\sqrt{x} \ln x + \frac{e}{\sqrt{x}}$
- 4614 $y = a + bx + cx^2 \sqrt{x} + d\sqrt{x} \ln x + e\frac{\ln x}{x}$
- 4615 $y = a + bx + cx^2 \sqrt{x} + d\sqrt{x} \ln x + \frac{e}{x}$
- 4616 $y = a + bx + cx^2 \sqrt{x} + d\sqrt{x} \ln x + \frac{e}{x\sqrt{x}}$
- 4617 $y = a + bx + cx^2 \sqrt{x} + d\sqrt{x} \ln x + e\frac{\ln x}{x^2}$
- 4618 $y = a + bx + cx^2 \sqrt{x} + d\sqrt{x} \ln x + \frac{e}{x^2}$

$$\begin{aligned}
 4619 \quad y &= a + bx + cx^2 \sqrt{x} + d\sqrt{x} \ln x + e e^{-x} \\
 4620 \quad y &= a + bx + cx^2 \sqrt{x} + d(\ln x)^2 + e \frac{x}{\ln x} \\
 4621 \quad y &= a + bx + cx^2 \sqrt{x} + d(\ln x)^2 + e\sqrt{x} \\
 4622 \quad y &= a + bx + cx^2 \sqrt{x} + d(\ln x)^2 + e \ln x \\
 4623 \quad y &= a + bx + cx^2 \sqrt{x} + d(\ln x)^2 + \frac{e}{\ln x} \\
 4624 \quad y &= a + bx + cx^2 \sqrt{x} + d(\ln x)^2 + \frac{e}{\sqrt{x}} \\
 4625 \quad y &= a + bx + cx^2 \sqrt{x} + d(\ln x)^2 + e \frac{\ln x}{x} \\
 4626 \quad y &= a + bx + cx^2 \sqrt{x} + d(\ln x)^2 + \frac{e}{x} \\
 4627 \quad y &= a + bx + cx^2 \sqrt{x} + d(\ln x)^2 + \frac{e}{x\sqrt{x}} \\
 4628 \quad y &= a + bx + cx^2 \sqrt{x} + d(\ln x)^2 + e \frac{\ln x}{x^2} \\
 4629 \quad y &= a + bx + cx^2 \sqrt{x} + d(\ln x)^2 + \frac{e}{x^2} \\
 4630 \quad y &= a + bx + cx^2 \sqrt{x} + d(\ln x)^2 + e e^{-x} \\
 4631 \quad y &= a + bx + cx^2 \sqrt{x} + d \frac{x}{\ln x} + e\sqrt{x} \\
 4632 \quad y &= a + bx + cx^2 \sqrt{x} + d \frac{x}{\ln x} + e \ln x \\
 4633 \quad y &= a + bx + cx^2 \sqrt{x} + d \frac{x}{\ln x} + \frac{e}{\ln x} \\
 4634 \quad y &= a + bx + cx^2 \sqrt{x} + d \frac{x}{\ln x} + \frac{e}{\sqrt{x}} \\
 4635 \quad y &= a + bx + cx^2 \sqrt{x} + d \frac{x}{\ln x} + e \frac{\ln x}{x} \\
 4636 \quad y &= a + bx + cx^2 \sqrt{x} + d \frac{x}{\ln x} + \frac{e}{x} \\
 4637 \quad y &= a + bx + cx^2 \sqrt{x} + d \frac{x}{\ln x} + \frac{e}{x\sqrt{x}} \\
 4638 \quad y &= a + bx + cx^2 \sqrt{x} + d \frac{x}{\ln x} + e \frac{\ln x}{x^2} \\
 4639 \quad y &= a + bx + cx^2 \sqrt{x} + d \frac{x}{\ln x} + \frac{e}{x^2} \\
 4640 \quad y &= a + bx + cx^2 \sqrt{x} + d \frac{x}{\ln x} + e e^{-x} \\
 4641 \quad y &= a + bx + cx^2 \sqrt{x} + d\sqrt{x} + e \ln x \\
 4642 \quad y &= a + bx + cx^2 \sqrt{x} + d\sqrt{x} + \frac{e}{\ln x} \\
 4643 \quad y &= a + bx + cx^2 \sqrt{x} + d\sqrt{x} + \frac{e}{\sqrt{x}} \\
 4644 \quad y &= a + bx + cx^2 \sqrt{x} + d\sqrt{x} + e \frac{\ln x}{x}
 \end{aligned}$$

$$\begin{aligned}
 4645 \quad y &= a + bx + cx^2 \sqrt{x} + d\sqrt{x} + \frac{e}{x} \\
 4646 \quad y &= a + bx + cx^2 \sqrt{x} + d\sqrt{x} + \frac{e}{x\sqrt{x}} \\
 4647 \quad y &= a + bx + cx^2 \sqrt{x} + d\sqrt{x} + e \frac{\ln x}{x^2} \\
 4648 \quad y &= a + bx + cx^2 \sqrt{x} + d\sqrt{x} + \frac{e}{x^2} \\
 4649 \quad y &= a + bx + cx^2 \sqrt{x} + d\sqrt{x} + e e^{-x} \\
 4650 \quad y &= a + bx + cx^2 \sqrt{x} + d \ln x + \frac{e}{\ln x} \\
 4651 \quad y &= a + bx + cx^2 \sqrt{x} + d \ln x + \frac{e}{\sqrt{x}} \\
 4652 \quad y &= a + bx + cx^2 \sqrt{x} + d \ln x + e \frac{\ln x}{x} \\
 4653 \quad y &= a + bx + cx^2 \sqrt{x} + d \ln x + \frac{e}{x} \\
 4654 \quad y &= a + bx + cx^2 \sqrt{x} + d \ln x + \frac{e}{x\sqrt{x}} \\
 4655 \quad y &= a + bx + cx^2 \sqrt{x} + d \ln x + e \frac{\ln x}{x^2} \\
 4656 \quad y &= a + bx + cx^2 \sqrt{x} + d \ln x + \frac{e}{x^2} \\
 4657 \quad y &= a + bx + cx^2 \sqrt{x} + d \ln x + e e^{-x} \\
 4658 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{\ln x} + \frac{e}{\sqrt{x}} \\
 4659 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{\ln x} + e \frac{\ln x}{x} \\
 4660 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{\ln x} + \frac{e}{x} \\
 4661 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{\ln x} + \frac{e}{x\sqrt{x}} \\
 4662 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{\ln x} + e \frac{\ln x}{x^2} \\
 4663 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{\ln x} + \frac{e}{x^2} \\
 4664 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{\ln x} + e e^{-x} \\
 4665 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x} \\
 4666 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{\sqrt{x}} + \frac{e}{x} \\
 4667 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{\sqrt{x}} + \frac{e}{x\sqrt{x}} \\
 4668 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x^2} \\
 4669 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{\sqrt{x}} + \frac{e}{x^2} \\
 4670 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{\sqrt{x}} + e e^{-x}
 \end{aligned}$$

$$\begin{aligned}
 4671 \quad y &= a + bx + cx^2 \sqrt{x} + d \frac{\ln x}{x} + \frac{e}{x} \\
 4672 \quad y &= a + bx + cx^2 \sqrt{x} + d \frac{\ln x}{x} + \frac{e}{x\sqrt{x}} \\
 4673 \quad y &= a + bx + cx^2 \sqrt{x} + d \frac{\ln x}{x} + e \frac{\ln x}{x^2} \\
 4674 \quad y &= a + bx + cx^2 \sqrt{x} + d \frac{\ln x}{x} + \frac{e}{x^2} \\
 4675 \quad y &= a + bx + cx^2 \sqrt{x} + d \frac{\ln x}{x} + e e^{-x} \\
 4676 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{x} + \frac{e}{x\sqrt{x}} \\
 4677 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{x} + e \frac{\ln x}{x^2} \\
 4678 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{x} + \frac{e}{x^2} \\
 4679 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{x} + e e^{-x} \\
 4680 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{x\sqrt{x}} + e \frac{\ln x}{x^2} \\
 4681 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{x\sqrt{x}} + \frac{e}{x^2} \\
 4682 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{x\sqrt{x}} + e e^{-x} \\
 4683 \quad y &= a + bx + cx^2 \sqrt{x} + d \frac{\ln x}{x^2} + \frac{e}{x^2} \\
 4684 \quad y &= a + bx + cx^2 \sqrt{x} + d \frac{\ln x}{x^2} + e e^{-x} \\
 4685 \quad y &= a + bx + cx^2 \sqrt{x} + \frac{d}{x^2} + e e^{-x} \\
 4686 \quad y &= a + bx + cx^3 + d e^x + e\sqrt{x} \ln x \\
 4687 \quad y &= a + bx + cx^3 + d e^x + e(\ln x)^2 \\
 4688 \quad y &= a + bx + cx^3 + d e^x + e \frac{x}{\ln x} \\
 4689 \quad y &= a + bx + cx^3 + d e^x + e\sqrt{x} \\
 4690 \quad y &= a + bx + cx^3 + d e^x + e \ln x \\
 4691 \quad y &= a + bx + cx^3 + d e^x + \frac{e}{\ln x} \\
 4692 \quad y &= a + bx + cx^3 + d e^x + \frac{e}{\sqrt{x}} \\
 4693 \quad y &= a + bx + cx^3 + d e^x + e \frac{\ln x}{x} \\
 4694 \quad y &= a + bx + cx^3 + d e^x + \frac{e}{x} \\
 4695 \quad y &= a + bx + cx^3 + d e^x + \frac{e}{x\sqrt{x}} \\
 4696 \quad y &= a + bx + cx^3 + d e^x + e \frac{\ln x}{x^2}
 \end{aligned}$$

$$4697 \ y = a + bx + cx^3 + d e^x + \frac{e}{x^2}$$

$$4698 \ y = a + bx + cx^3 + d e^x + e e^{-x}$$

$$4699 \ y = a + bx + cx^3 + d\sqrt{x} \ln x + e(\ln x)^2$$

$$4700 \ y = a + bx + cx^3 + d\sqrt{x} \ln x + e \frac{x}{\ln x}$$

$$4701 \ y = a + bx + cx^3 + d\sqrt{x} \ln x + e\sqrt{x}$$

$$4702 \ y = a + bx + cx^3 + d\sqrt{x} \ln x + e \ln x$$

$$4703 \ y = a + bx + cx^3 + d\sqrt{x} \ln x + e \frac{e}{\ln x}$$

$$4704 \ y = a + bx + cx^3 + d\sqrt{x} \ln x + e \frac{e}{\sqrt{x}}$$

$$4705 \ y = a + bx + cx^3 + d\sqrt{x} \ln x + e \frac{\ln x}{x}$$

$$4706 \ y = a + bx + cx^3 + d\sqrt{x} \ln x + e \frac{e}{x}$$

$$4707 \ y = a + bx + cx^3 + d\sqrt{x} \ln x + e \frac{e}{x\sqrt{x}}$$

$$4708 \ y = a + bx + cx^3 + d\sqrt{x} \ln x + e \frac{\ln x}{x^2}$$

$$4709 \ y = a + bx + cx^3 + d\sqrt{x} \ln x + e \frac{e}{x^2}$$

$$4710 \ y = a + bx + cx^3 + d\sqrt{x} \ln x + e e^{-x}$$

$$4711 \ y = a + bx + cx^3 + d(\ln x)^2 + e \frac{x}{\ln x}$$

$$4712 \ y = a + bx + cx^3 + d(\ln x)^2 + e\sqrt{x}$$

$$4713 \ y = a + bx + cx^3 + d(\ln x)^2 + e \ln x$$

$$4714 \ y = a + bx + cx^3 + d(\ln x)^2 + e \frac{e}{\ln x}$$

$$4715 \ y = a + bx + cx^3 + d(\ln x)^2 + e \frac{e}{\sqrt{x}}$$

$$4716 \ y = a + bx + cx^3 + d(\ln x)^2 + e \frac{\ln x}{x}$$

$$4717 \ y = a + bx + cx^3 + d(\ln x)^2 + e \frac{e}{x}$$

$$4718 \ y = a + bx + cx^3 + d(\ln x)^2 + e \frac{e}{x\sqrt{x}}$$

$$4719 \ y = a + bx + cx^3 + d(\ln x)^2 + e \frac{\ln x}{x^2}$$

$$4720 \ y = a + bx + cx^3 + d(\ln x)^2 + e \frac{e}{x^2}$$

$$4721 \ y = a + bx + cx^3 + d(\ln x)^2 + e e^{-x}$$

$$4722 \ y = a + bx + cx^3 + d \frac{x}{\ln x} + e\sqrt{x}$$

$$4723 \ y = a + bx + cx^3 + d \frac{x}{\ln x} + e \ln x$$

$$4724 \ y = a + bx + cx^3 + d \frac{x}{\ln x} + e \frac{e}{\ln x}$$

$$4725 \ y = a + bx + cx^3 + d \frac{x}{\ln x} + e \frac{e}{\sqrt{x}}$$

$$4726 \ y = a + bx + cx^3 + d \frac{x}{\ln x} + e \frac{\ln x}{x}$$

$$4727 \ y = a + bx + cx^3 + d \frac{x}{\ln x} + e \frac{e}{x}$$

$$4728 \ y = a + bx + cx^3 + d \frac{x}{\ln x} + e \frac{e}{x\sqrt{x}}$$

$$4729 \ y = a + bx + cx^3 + d \frac{x}{\ln x} + e \frac{\ln x}{x^2}$$

$$4730 \ y = a + bx + cx^3 + d \frac{x}{\ln x} + e \frac{e}{x^2}$$

$$4731 \ y = a + bx + cx^3 + d \frac{x}{\ln x} + e e^{-x}$$

$$4732 \ y = a + bx + cx^3 + d\sqrt{x} + e \ln x$$

$$4733 \ y = a + bx + cx^3 + d\sqrt{x} + e \frac{e}{\ln x}$$

$$4734 \ y = a + bx + cx^3 + d\sqrt{x} + e \frac{e}{\sqrt{x}}$$

$$4735 \ y = a + bx + cx^3 + d\sqrt{x} + e \frac{\ln x}{x}$$

$$4736 \ y = a + bx + cx^3 + d\sqrt{x} + e \frac{e}{x}$$

$$4737 \ y = a + bx + cx^3 + d\sqrt{x} + e \frac{e}{x\sqrt{x}}$$

$$4738 \ y = a + bx + cx^3 + d\sqrt{x} + e \frac{\ln x}{x^2}$$

$$4739 \ y = a + bx + cx^3 + d\sqrt{x} + e \frac{e}{x^2}$$

$$4740 \ y = a + bx + cx^3 + d\sqrt{x} + e e^{-x}$$

$$4741 \ y = a + bx + cx^3 + d \ln x + e \frac{e}{\ln x}$$

$$4742 \ y = a + bx + cx^3 + d \ln x + e \frac{e}{\sqrt{x}}$$

$$4743 \ y = a + bx + cx^3 + d \ln x + e \frac{\ln x}{x}$$

$$4744 \ y = a + bx + cx^3 + d \ln x + e \frac{e}{x}$$

$$4745 \ y = a + bx + cx^3 + d \ln x + e \frac{e}{x\sqrt{x}}$$

$$4746 \ y = a + bx + cx^3 + d \ln x + e \frac{\ln x}{x^2}$$

$$4747 \ y = a + bx + cx^3 + d \ln x + e \frac{e}{x^2}$$

$$4748 \ y = a + bx + cx^3 + d \ln x + e e^{-x}$$

$$4749 \ y = a + bx + cx^3 + \frac{d}{\ln x} + \frac{e}{\sqrt{x}}$$

$$4750 \ y = a + bx + cx^3 + \frac{d}{\ln x} + e \frac{\ln x}{x}$$

$$4751 \ y = a + bx + cx^3 + \frac{d}{\ln x} + e \frac{e}{x}$$

$$4752 \ y = a + bx + cx^3 + \frac{d}{\ln x} + e \frac{e}{x\sqrt{x}}$$

$$4753 \ y = a + bx + cx^3 + \frac{d}{\ln x} + e \frac{\ln x}{x^2}$$

$$4754 \ y = a + bx + cx^3 + \frac{d}{\ln x} + e \frac{e}{x^2}$$

$$4755 \ y = a + bx + cx^3 + \frac{d}{\ln x} + e e^{-x}$$

$$4756 \ y = a + bx + cx^3 + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x}$$

$$4757 \ y = a + bx + cx^3 + \frac{d}{\sqrt{x}} + e \frac{e}{x}$$

$$4758 \ y = a + bx + cx^3 + \frac{d}{\sqrt{x}} + e \frac{e}{x\sqrt{x}}$$

$$4759 \ y = a + bx + cx^3 + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x^2}$$

$$4760 \ y = a + bx + cx^3 + \frac{d}{\sqrt{x}} + e \frac{e}{x^2}$$

$$4761 \ y = a + bx + cx^3 + \frac{d}{\sqrt{x}} + e e^{-x}$$

$$4762 \ y = a + bx + cx^3 + d \frac{\ln x}{x} + e \frac{e}{x}$$

$$4763 \ y = a + bx + cx^3 + d \frac{\ln x}{x} + e \frac{e}{x\sqrt{x}}$$

$$4764 \ y = a + bx + cx^3 + d \frac{\ln x}{x} + e \frac{\ln x}{x^2}$$

$$4765 \ y = a + bx + cx^3 + d \frac{\ln x}{x} + e \frac{e}{x^2}$$

$$4766 \ y = a + bx + cx^3 + d \frac{\ln x}{x} + e e^{-x}$$

$$4767 \ y = a + bx + cx^3 + \frac{d}{x} + e \frac{e}{x\sqrt{x}}$$

$$4768 \ y = a + bx + cx^3 + \frac{d}{x} + e \frac{\ln x}{x^2}$$

$$4769 \ y = a + bx + cx^3 + \frac{d}{x} + e \frac{e}{x^2}$$

$$4770 \ y = a + bx + cx^3 + \frac{d}{x} + e e^{-x}$$

$$4771 \ y = a + bx + cx^3 + \frac{d}{x\sqrt{x}} + e \frac{\ln x}{x^2}$$

$$4772 \ y = a + bx + cx^3 + \frac{d}{x\sqrt{x}} + e \frac{e}{x^2}$$

$$4773 \ y = a + bx + cx^3 + \frac{d}{x\sqrt{x}} + e e^{-x}$$

$$4774 \ y = a + bx + cx^3 + d \frac{\ln x}{x^2} + e \frac{e}{x^2}$$

- 4775 $y = a + bx + cx^3 + d \frac{\ln x}{x^2} + e e^{-x}$
- 4776 $y = a + bx + cx^3 + \frac{d}{x^2} + e e^{-x}$
- 4777 $y = a + bx + c e^x + d \sqrt{x} \ln x + e (\ln x)^2$
- 4778 $y = a + bx + c e^x + d \sqrt{x} \ln x + e \frac{x}{\ln x}$
- 4779 $y = a + bx + c e^x + d \sqrt{x} \ln x + e \sqrt{x}$
- 4780 $y = a + bx + c e^x + d \sqrt{x} \ln x + e \ln x$
- 4781 $y = a + bx + c e^x + d \sqrt{x} \ln x + \frac{e}{\ln x}$
- 4782 $y = a + bx + c e^x + d \sqrt{x} \ln x + \frac{e}{\sqrt{x}}$
- 4783 $y = a + bx + c e^x + d \sqrt{x} \ln x + e \frac{\ln x}{x}$
- 4784 $y = a + bx + c e^x + d \sqrt{x} \ln x + \frac{e}{x}$
- 4785 $y = a + bx + c e^x + d \sqrt{x} \ln x + \frac{e}{x \sqrt{x}}$
- 4786 $y = a + bx + c e^x + d \sqrt{x} \ln x + e \frac{\ln x}{x^2}$
- 4787 $y = a + bx + c e^x + d \sqrt{x} \ln x + \frac{e}{x^2}$
- 4788 $y = a + bx + c e^x + d \sqrt{x} \ln x + e e^{-x}$
- 4789 $y = a + bx + c e^x + d (\ln x)^2 + e \frac{x}{\ln x}$
- 4790 $y = a + bx + c e^x + d (\ln x)^2 + e \sqrt{x}$
- 4791 $y = a + bx + c e^x + d (\ln x)^2 + e \ln x$
- 4792 $y = a + bx + c e^x + d (\ln x)^2 + \frac{e}{\ln x}$
- 4793 $y = a + bx + c e^x + d (\ln x)^2 + \frac{e}{\sqrt{x}}$
- 4794 $y = a + bx + c e^x + d (\ln x)^2 + e \frac{\ln x}{x}$
- 4795 $y = a + bx + c e^x + d (\ln x)^2 + \frac{e}{x}$
- 4796 $y = a + bx + c e^x + d (\ln x)^2 + \frac{e}{x \sqrt{x}}$
- 4797 $y = a + bx + c e^x + d (\ln x)^2 + e \frac{\ln x}{x^2}$
- 4798 $y = a + bx + c e^x + d (\ln x)^2 + \frac{e}{x^2}$
- 4799 $y = a + bx + c e^x + d (\ln x)^2 + e e^{-x}$
- 4800 $y = a + bx + c e^x + d \frac{x}{\ln x} + e \sqrt{x}$
- 4801 $y = a + bx + c e^x + d \frac{x}{\ln x} + e \ln x$
- 4802 $y = a + bx + c e^x + d \frac{x}{\ln x} + \frac{e}{\ln x}$
- 4803 $y = a + bx + c e^x + d \frac{x}{\ln x} + \frac{e}{\sqrt{x}}$
- 4804 $y = a + bx + c e^x + d \frac{x}{\ln x} + e \frac{\ln x}{x}$
- 4805 $y = a + bx + c e^x + d \frac{x}{\ln x} + \frac{e}{x}$
- 4806 $y = a + bx + c e^x + d \frac{x}{\ln x} + \frac{e}{x \sqrt{x}}$
- 4807 $y = a + bx + c e^x + d \frac{x}{\ln x} + e \frac{\ln x}{x^2}$
- 4808 $y = a + bx + c e^x + d \frac{x}{\ln x} + \frac{e}{x^2}$
- 4809 $y = a + bx + c e^x + d \frac{x}{\ln x} + e e^{-x}$
- 4810 $y = a + bx + c e^x + d \sqrt{x} + e \ln x$
- 4811 $y = a + bx + c e^x + d \sqrt{x} + \frac{e}{\ln x}$
- 4812 $y = a + bx + c e^x + d \sqrt{x} + \frac{e}{\sqrt{x}}$
- 4813 $y = a + bx + c e^x + d \sqrt{x} + e \frac{\ln x}{x}$
- 4814 $y = a + bx + c e^x + d \sqrt{x} + \frac{e}{x}$
- 4815 $y = a + bx + c e^x + d \sqrt{x} + \frac{e}{x \sqrt{x}}$
- 4816 $y = a + bx + c e^x + d \sqrt{x} + e \frac{\ln x}{x^2}$
- 4817 $y = a + bx + c e^x + d \sqrt{x} + \frac{e}{x^2}$
- 4818 $y = a + bx + c e^x + d \sqrt{x} + e e^{-x}$
- 4819 $y = a + bx + c e^x + d \ln x + \frac{e}{\ln x}$
- 4820 $y = a + bx + c e^x + d \ln x + \frac{e}{\sqrt{x}}$
- 4821 $y = a + bx + c e^x + d \ln x + e \frac{\ln x}{x}$
- 4822 $y = a + bx + c e^x + d \ln x + \frac{e}{x}$
- 4823 $y = a + bx + c e^x + d \ln x + \frac{e}{x \sqrt{x}}$
- 4824 $y = a + bx + c e^x + d \ln x + e \frac{\ln x}{x^2}$
- 4825 $y = a + bx + c e^x + d \ln x + \frac{e}{x^2}$
- 4826 $y = a + bx + c e^x + d \ln x + e e^{-x}$
- 4827 $y = a + bx + c e^x + \frac{d}{\ln x} + \frac{e}{\sqrt{x}}$
- 4828 $y = a + bx + c e^x + \frac{d}{\ln x} + e \frac{\ln x}{x}$
- 4829 $y = a + bx + c e^x + \frac{d}{\ln x} + \frac{e}{x}$
- 4830 $y = a + bx + c e^x + \frac{d}{\ln x} + \frac{e}{x \sqrt{x}}$
- 4831 $y = a + bx + c e^x + \frac{d}{\ln x} + e \frac{\ln x}{x^2}$
- 4832 $y = a + bx + c e^x + \frac{d}{\ln x} + \frac{e}{x^2}$
- 4833 $y = a + bx + c e^x + \frac{d}{\ln x} + e e^{-x}$
- 4834 $y = a + bx + c e^x + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x}$
- 4835 $y = a + bx + c e^x + \frac{d}{\sqrt{x}} + \frac{e}{x}$
- 4836 $y = a + bx + c e^x + \frac{d}{\sqrt{x}} + \frac{e}{x \sqrt{x}}$
- 4837 $y = a + bx + c e^x + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x^2}$
- 4838 $y = a + bx + c e^x + \frac{d}{\sqrt{x}} + \frac{e}{x^2}$
- 4839 $y = a + bx + c e^x + \frac{d}{\sqrt{x}} + e e^{-x}$
- 4840 $y = a + bx + c e^x + d \frac{\ln x}{x} + \frac{e}{x}$
- 4841 $y = a + bx + c e^x + d \frac{\ln x}{x} + \frac{e}{x \sqrt{x}}$
- 4842 $y = a + bx + c e^x + d \frac{\ln x}{x} + e \frac{\ln x}{x^2}$
- 4843 $y = a + bx + c e^x + d \frac{\ln x}{x} + \frac{e}{x^2}$
- 4844 $y = a + bx + c e^x + d \frac{\ln x}{x} + e e^{-x}$
- 4845 $y = a + bx + c e^x + \frac{d}{x} + \frac{e}{x \sqrt{x}}$
- 4846 $y = a + bx + c e^x + \frac{d}{x} + e \frac{\ln x}{x^2}$
- 4847 $y = a + bx + c e^x + \frac{d}{x} + \frac{e}{x^2}$
- 4848 $y = a + bx + c e^x + \frac{d}{x} + e e^{-x}$
- 4849 $y = a + bx + c e^x + \frac{d}{x \sqrt{x}} + e \frac{\ln x}{x^2}$
- 4850 $y = a + bx + c e^x + \frac{d}{x \sqrt{x}} + \frac{e}{x^2}$
- 4851 $y = a + bx + c e^x + \frac{d}{x \sqrt{x}} + e e^{-x}$
- 4852 $y = a + bx + c e^x + d \frac{\ln x}{x^2} + \frac{e}{x^2}$

- 4853 $y = a + bx + ce^x + d \frac{\ln x}{x^2} + e e^{-x}$
- 4854 $y = a + bx + ce^x + \frac{d}{x^2} + e e^{-x}$
- 4855 $y = a + bx + c\sqrt{x} \ln x + d(\ln x)^2 + e \frac{x}{\ln x}$
- 4856 $y = a + bx + c\sqrt{x} \ln x + d(\ln x)^2 + e\sqrt{x}$
- 4857 $y = a + bx + c\sqrt{x} \ln x + d(\ln x)^2 + e \ln x$
- 4858 $y = a + bx + c\sqrt{x} \ln x + d(\ln x)^2 + \frac{e}{\ln x}$
- 4859 $y = a + bx + c\sqrt{x} \ln x + d(\ln x)^2 + \frac{e}{\sqrt{x}}$
- 4860 $y = a + bx + c\sqrt{x} \ln x + d(\ln x)^2 + e \frac{\ln x}{x}$
- 4861 $y = a + bx + c\sqrt{x} \ln x + d(\ln x)^2 + \frac{e}{x}$
- 4862 $y = a + bx + c\sqrt{x} \ln x + d(\ln x)^2 + \frac{e}{x\sqrt{x}}$
- 4863 $y = a + bx + c\sqrt{x} \ln x + d(\ln x)^2 + e \frac{\ln x}{x^2}$
- 4864 $y = a + bx + c\sqrt{x} \ln x + d(\ln x)^2 + \frac{e}{x^2}$
- 4865 $y = a + bx + c\sqrt{x} \ln x + d(\ln x)^2 + e e^{-x}$
- 4866 $y = a + bx + c\sqrt{x} \ln x + d \frac{x}{\ln x} + e\sqrt{x}$
- 4867 $y = a + bx + c\sqrt{x} \ln x + d \frac{x}{\ln x} + e \ln x$
- 4868 $y = a + bx + c\sqrt{x} \ln x + d \frac{x}{\ln x} + \frac{e}{\ln x}$
- 4869 $y = a + bx + c\sqrt{x} \ln x + d \frac{x}{\ln x} + \frac{e}{\sqrt{x}}$
- 4870 $y = a + bx + c\sqrt{x} \ln x + d \frac{x}{\ln x} + e \frac{\ln x}{x}$
- 4871 $y = a + bx + c\sqrt{x} \ln x + d \frac{x}{\ln x} + \frac{e}{x}$
- 4872 $y = a + bx + c\sqrt{x} \ln x + d \frac{x}{\ln x} + \frac{e}{x\sqrt{x}}$
- 4873 $y = a + bx + c\sqrt{x} \ln x + d \frac{x}{\ln x} + e \frac{\ln x}{x^2}$
- 4874 $y = a + bx + c\sqrt{x} \ln x + d \frac{x}{\ln x} + \frac{e}{x^2}$
- 4875 $y = a + bx + c\sqrt{x} \ln x + d \frac{x}{\ln x} + e e^{-x}$
- 4876 $y = a + bx + c\sqrt{x} \ln x + d\sqrt{x} + e \ln x$
- 4877 $y = a + bx + c\sqrt{x} \ln x + d\sqrt{x} + \frac{e}{\ln x}$
- 4878 $y = a + bx + c\sqrt{x} \ln x + d\sqrt{x} + \frac{e}{\sqrt{x}}$
- 4879 $y = a + bx + c\sqrt{x} \ln x + d\sqrt{x} + e \frac{\ln x}{x}$
- 4880 $y = a + bx + c\sqrt{x} \ln x + d\sqrt{x} + \frac{e}{x}$
- 4881 $y = a + bx + c\sqrt{x} \ln x + d\sqrt{x} + \frac{e}{x\sqrt{x}}$
- 4882 $y = a + bx + c\sqrt{x} \ln x + d\sqrt{x} + e \frac{\ln x}{x^2}$
- 4883 $y = a + bx + c\sqrt{x} \ln x + d\sqrt{x} + \frac{e}{x^2}$
- 4884 $y = a + bx + c\sqrt{x} \ln x + d\sqrt{x} + e e^{-x}$
- 4885 $y = a + bx + c\sqrt{x} \ln x + d \ln x + \frac{e}{\ln x}$
- 4886 $y = a + bx + c\sqrt{x} \ln x + d \ln x + \frac{e}{\sqrt{x}}$
- 4887 $y = a + bx + c\sqrt{x} \ln x + d \ln x + e \frac{\ln x}{x}$
- 4888 $y = a + bx + c\sqrt{x} \ln x + d \ln x + \frac{e}{x}$
- 4889 $y = a + bx + c\sqrt{x} \ln x + d \ln x + \frac{e}{x\sqrt{x}}$
- 4890 $y = a + bx + c\sqrt{x} \ln x + d \ln x + e \frac{\ln x}{x^2}$
- 4891 $y = a + bx + c\sqrt{x} \ln x + d \ln x + \frac{e}{x^2}$
- 4892 $y = a + bx + c\sqrt{x} \ln x + d \ln x + e e^{-x}$
- 4893 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{\ln x} + \frac{e}{\sqrt{x}}$
- 4894 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{\ln x} + e \frac{\ln x}{x}$
- 4895 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{\ln x} + \frac{e}{x}$
- 4896 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{\ln x} + \frac{e}{x\sqrt{x}}$
- 4897 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{\ln x} + e \frac{\ln x}{x^2}$
- 4898 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{\ln x} + \frac{e}{x^2}$
- 4899 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{\ln x} + e e^{-x}$
- 4900 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x}$
- 4901 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{\sqrt{x}} + \frac{e}{x}$
- 4902 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{\sqrt{x}} + \frac{e}{x\sqrt{x}}$
- 4903 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x^2}$
- 4904 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{\sqrt{x}} + \frac{e}{x^2}$
- 4905 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{\sqrt{x}} + e e^{-x}$
- 4906 $y = a + bx + c\sqrt{x} \ln x + d \frac{\ln x}{x} + \frac{e}{x}$
- 4907 $y = a + bx + c\sqrt{x} \ln x + d \frac{\ln x}{x} + \frac{e}{x\sqrt{x}}$
- 4908 $y = a + bx + c\sqrt{x} \ln x + d \frac{\ln x}{x} + e \frac{\ln x}{x^2}$
- 4909 $y = a + bx + c\sqrt{x} \ln x + d \frac{\ln x}{x} + \frac{e}{x^2}$
- 4910 $y = a + bx + c\sqrt{x} \ln x + d \frac{\ln x}{x} + e e^{-x}$
- 4911 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{x} + \frac{e}{x\sqrt{x}}$
- 4912 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{x} + e \frac{\ln x}{x^2}$
- 4913 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{x} + \frac{e}{x^2}$
- 4914 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{x} + e e^{-x}$
- 4915 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{x\sqrt{x}} + e \frac{\ln x}{x^2}$
- 4916 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{x\sqrt{x}} + \frac{e}{x^2}$
- 4917 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{x\sqrt{x}} + e e^{-x}$
- 4918 $y = a + bx + c\sqrt{x} \ln x + d \frac{\ln x}{x^2} + \frac{e}{x^2}$
- 4919 $y = a + bx + c\sqrt{x} \ln x + d \frac{\ln x}{x^2} + e e^{-x}$
- 4920 $y = a + bx + c\sqrt{x} \ln x + \frac{d}{x^2} + e e^{-x}$
- 4921 $y = a + bx + c(\ln x)^2 + d \frac{x}{\ln x} + e\sqrt{x}$
- 4922 $y = a + bx + c(\ln x)^2 + d \frac{x}{\ln x} + e \ln x$
- 4923 $y = a + bx + c(\ln x)^2 + d \frac{x}{\ln x} + \frac{e}{\ln x}$
- 4924 $y = a + bx + c(\ln x)^2 + d \frac{x}{\ln x} + \frac{e}{\sqrt{x}}$
- 4925 $y = a + bx + c(\ln x)^2 + d \frac{x}{\ln x} + e \frac{\ln x}{x}$
- 4926 $y = a + bx + c(\ln x)^2 + d \frac{x}{\ln x} + \frac{e}{x}$
- 4927 $y = a + bx + c(\ln x)^2 + d \frac{x}{\ln x} + \frac{e}{x\sqrt{x}}$
- 4928 $y = a + bx + c(\ln x)^2 + d \frac{x}{\ln x} + e \frac{\ln x}{x^2}$
- 4929 $y = a + bx + c(\ln x)^2 + d \frac{x}{\ln x} + \frac{e}{x^2}$
- 4930 $y = a + bx + c(\ln x)^2 + d \frac{x}{\ln x} + e e^{-x}$

- 4931 $y = a + bx + c(\ln x)^2 + d\sqrt{x} + e \ln x$
- 4932 $y = a + bx + c(\ln x)^2 + d\sqrt{x} + \frac{e}{\ln x}$
- 4933 $y = a + bx + c(\ln x)^2 + d\sqrt{x} + \frac{e}{\sqrt{x}}$
- 4934 $y = a + bx + c(\ln x)^2 + d\sqrt{x} + e \frac{\ln x}{x}$
- 4935 $y = a + bx + c(\ln x)^2 + d\sqrt{x} + \frac{e}{x}$
- 4936 $y = a + bx + c(\ln x)^2 + d\sqrt{x} + \frac{e}{x\sqrt{x}}$
- 4937 $y = a + bx + c(\ln x)^2 + d\sqrt{x} + e \frac{\ln x}{x^2}$
- 4938 $y = a + bx + c(\ln x)^2 + d\sqrt{x} + \frac{e}{x^2}$
- 4939 $y = a + bx + c(\ln x)^2 + d\sqrt{x} + e e^{-x}$
- 4940 $y = a + bx + c(\ln x)^2 + d \ln x + \frac{e}{\ln x}$
- 4941 $y = a + bx + c(\ln x)^2 + d \ln x + \frac{e}{\sqrt{x}}$
- 4942 $y = a + bx + c(\ln x)^2 + d \ln x + e \frac{\ln x}{x}$
- 4943 $y = a + bx + c(\ln x)^2 + d \ln x + \frac{e}{x}$
- 4944 $y = a + bx + c(\ln x)^2 + d \ln x + \frac{e}{x\sqrt{x}}$
- 4945 $y = a + bx + c(\ln x)^2 + d \ln x + e \frac{\ln x}{x^2}$
- 4946 $y = a + bx + c(\ln x)^2 + d \ln x + \frac{e}{x^2}$
- 4947 $y = a + bx + c(\ln x)^2 + d \ln x + e e^{-x}$
- 4948 $y = a + bx + c(\ln x)^2 + \frac{d}{\ln x} + \frac{e}{\sqrt{x}}$
- 4949 $y = a + bx + c(\ln x)^2 + \frac{d}{\ln x} + e \frac{\ln x}{x}$
- 4950 $y = a + bx + c(\ln x)^2 + \frac{d}{\ln x} + \frac{e}{x}$
- 4951 $y = a + bx + c(\ln x)^2 + \frac{d}{\ln x} + \frac{e}{x\sqrt{x}}$
- 4952 $y = a + bx + c(\ln x)^2 + \frac{d}{\ln x} + e \frac{\ln x}{x^2}$
- 4953 $y = a + bx + c(\ln x)^2 + \frac{d}{\ln x} + \frac{e}{x^2}$
- 4954 $y = a + bx + c(\ln x)^2 + \frac{d}{\ln x} + e e^{-x}$
- 4955 $y = a + bx + c(\ln x)^2 + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x}$
- 4956 $y = a + bx + c(\ln x)^2 + \frac{d}{\sqrt{x}} + \frac{e}{x}$
- 4957 $y = a + bx + c(\ln x)^2 + \frac{d}{\sqrt{x}} + \frac{e}{x\sqrt{x}}$
- 4958 $y = a + bx + c(\ln x)^2 + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x^2}$
- 4959 $y = a + bx + c(\ln x)^2 + \frac{d}{\sqrt{x}} + \frac{e}{x^2}$
- 4960 $y = a + bx + c(\ln x)^2 + \frac{d}{\sqrt{x}} + e e^{-x}$
- 4961 $y = a + bx + c(\ln x)^2 + d \frac{\ln x}{x} + \frac{e}{x}$
- 4962 $y = a + bx + c(\ln x)^2 + d \frac{\ln x}{x} + \frac{e}{x\sqrt{x}}$
- 4963 $y = a + bx + c(\ln x)^2 + d \frac{\ln x}{x} + e \frac{\ln x}{x^2}$
- 4964 $y = a + bx + c(\ln x)^2 + d \frac{\ln x}{x} + \frac{e}{x^2}$
- 4965 $y = a + bx + c(\ln x)^2 + d \frac{\ln x}{x} + e e^{-x}$
- 4966 $y = a + bx + c(\ln x)^2 + \frac{d}{x} + \frac{e}{x\sqrt{x}}$
- 4967 $y = a + bx + c(\ln x)^2 + \frac{d}{x} + e \frac{\ln x}{x^2}$
- 4968 $y = a + bx + c(\ln x)^2 + \frac{d}{x} + \frac{e}{x^2}$
- 4969 $y = a + bx + c(\ln x)^2 + \frac{d}{x} + e e^{-x}$
- 4970 $y = a + bx + c(\ln x)^2 + \frac{d}{x\sqrt{x}} + e \frac{\ln x}{x^2}$
- 4971 $y = a + bx + c(\ln x)^2 + \frac{d}{x\sqrt{x}} + \frac{e}{x^2}$
- 4972 $y = a + bx + c(\ln x)^2 + \frac{d}{x\sqrt{x}} + e e^{-x}$
- 4973 $y = a + bx + c(\ln x)^2 + d \frac{\ln x}{x^2} + \frac{e}{x^2}$
- 4974 $y = a + bx + c(\ln x)^2 + d \frac{\ln x}{x^2} + e e^{-x}$
- 4975 $y = a + bx + c(\ln x)^2 + \frac{d}{x^2} + e e^{-x}$
- 4976 $y = a + bx + c \frac{x}{\ln x} + d\sqrt{x} + e \ln x$
- 4977 $y = a + bx + c \frac{x}{\ln x} + d\sqrt{x} + \frac{e}{\ln x}$
- 4978 $y = a + bx + c \frac{x}{\ln x} + d\sqrt{x} + \frac{e}{\sqrt{x}}$
- 4979 $y = a + bx + c \frac{x}{\ln x} + d\sqrt{x} + e \frac{\ln x}{x}$
- 4980 $y = a + bx + c \frac{x}{\ln x} + d\sqrt{x} + \frac{e}{x}$
- 4981 $y = a + bx + c \frac{x}{\ln x} + d\sqrt{x} + \frac{e}{x\sqrt{x}}$
- 4982 $y = a + bx + c \frac{x}{\ln x} + d\sqrt{x} + e \frac{\ln x}{x^2}$
- 4983 $y = a + bx + c \frac{x}{\ln x} + d\sqrt{x} + \frac{e}{x^2}$
- 4984 $y = a + bx + c \frac{x}{\ln x} + d\sqrt{x} + e e^{-x}$
- 4985 $y = a + bx + c \frac{x}{\ln x} + d \ln x + \frac{e}{\ln x}$
- 4986 $y = a + bx + c \frac{x}{\ln x} + d \ln x + \frac{e}{\sqrt{x}}$
- 4987 $y = a + bx + c \frac{x}{\ln x} + d \ln x + e \frac{\ln x}{x}$
- 4988 $y = a + bx + c \frac{x}{\ln x} + d \ln x + \frac{e}{x}$
- 4989 $y = a + bx + c \frac{x}{\ln x} + d \ln x + \frac{e}{x\sqrt{x}}$
- 4990 $y = a + bx + c \frac{x}{\ln x} + d \ln x + e \frac{\ln x}{x^2}$
- 4991 $y = a + bx + c \frac{x}{\ln x} + d \ln x + \frac{e}{x^2}$
- 4992 $y = a + bx + c \frac{x}{\ln x} + d \ln x + e e^{-x}$
- 4993 $y = a + bx + c \frac{x}{\ln x} + \frac{d}{\ln x} + \frac{e}{\sqrt{x}}$
- 4994 $y = a + bx + c \frac{x}{\ln x} + \frac{d}{\ln x} + e \frac{\ln x}{x}$
- 4995 $y = a + bx + c \frac{x}{\ln x} + \frac{d}{\ln x} + \frac{e}{x}$
- 4996 $y = a + bx + c \frac{x}{\ln x} + \frac{d}{\ln x} + \frac{e}{x\sqrt{x}}$
- 4997 $y = a + bx + c \frac{x}{\ln x} + \frac{d}{\ln x} + e \frac{\ln x}{x^2}$
- 4998 $y = a + bx + c \frac{x}{\ln x} + \frac{d}{\ln x} + \frac{e}{x^2}$
- 4999 $y = a + bx + c \frac{x}{\ln x} + \frac{d}{\ln x} + e e^{-x}$
- 5000 $y = a + bx + c \frac{x}{\ln x} + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x}$
- 5001 $y = a + bx + c \frac{x}{\ln x} + \frac{d}{\sqrt{x}} + \frac{e}{x}$
- 5002 $y = a + bx + c \frac{x}{\ln x} + \frac{d}{\sqrt{x}} + \frac{e}{x\sqrt{x}}$
- 5003 $y = a + bx + c \frac{x}{\ln x} + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x^2}$
- 5004 $y = a + bx + c \frac{x}{\ln x} + \frac{d}{\sqrt{x}} + \frac{e}{x^2}$
- 5005 $y = a + bx + c \frac{x}{\ln x} + \frac{d}{\sqrt{x}} + e e^{-x}$
- 5006 $y = a + bx + c \frac{x}{\ln x} + d \frac{\ln x}{x} + \frac{e}{x}$
- 5007 $y = a + bx + c \frac{x}{\ln x} + d \frac{\ln x}{x} + \frac{e}{x\sqrt{x}}$
- 5008 $y = a + bx + c \frac{x}{\ln x} + d \frac{\ln x}{x} + e \frac{\ln x}{x^2}$

$$\begin{aligned}
5009 \quad y &= a + bx + c \frac{x}{\ln x} + d \frac{\ln x}{x} + \frac{e}{x^2} \\
5010 \quad y &= a + bx + c \frac{x}{\ln x} + d \frac{\ln x}{x} + e e^{-x} \\
5011 \quad y &= a + bx + c \frac{x}{\ln x} + \frac{d}{x} + \frac{e}{x\sqrt{x}} \\
5012 \quad y &= a + bx + c \frac{x}{\ln x} + \frac{d}{x} + e \frac{\ln x}{x^2} \\
5013 \quad y &= a + bx + c \frac{x}{\ln x} + \frac{d}{x} + \frac{e}{x^2} \\
5014 \quad y &= a + bx + c \frac{x}{\ln x} + \frac{d}{x} + e e^{-x} \\
5015 \quad y &= a + bx + c \frac{x}{\ln x} + \frac{d}{x\sqrt{x}} + e \frac{\ln x}{x^2} \\
5016 \quad y &= a + bx + c \frac{x}{\ln x} + \frac{d}{x\sqrt{x}} + \frac{e}{x^2} \\
5017 \quad y &= a + bx + c \frac{x}{\ln x} + \frac{d}{x\sqrt{x}} + e e^{-x} \\
5018 \quad y &= a + bx + c \frac{x}{\ln x} + d \frac{\ln x}{x^2} + \frac{e}{x^2} \\
5019 \quad y &= a + bx + c \frac{x}{\ln x} + d \frac{\ln x}{x^2} + e e^{-x} \\
5020 \quad y &= a + bx + c \frac{x}{\ln x} + \frac{d}{x^2} + e e^{-x} \\
5021 \quad y &= a + bx + c\sqrt{x} + d \ln x + \frac{e}{\ln x} \\
5022 \quad y &= a + bx + c\sqrt{x} + d \ln x + \frac{e}{\sqrt{x}} \\
5023 \quad y &= a + bx + c\sqrt{x} + d \ln x + e \frac{\ln x}{x} \\
5024 \quad y &= a + bx + c\sqrt{x} + d \ln x + \frac{e}{x} \\
5025 \quad y &= a + bx + c\sqrt{x} + d \ln x + \frac{e}{x\sqrt{x}} \\
5026 \quad y &= a + bx + c\sqrt{x} + d \ln x + e \frac{\ln x}{x^2} \\
5027 \quad y &= a + bx + c\sqrt{x} + d \ln x + \frac{e}{x^2} \\
5028 \quad y &= a + bx + c\sqrt{x} + d \ln x + e e^{-x} \\
5029 \quad y &= a + bx + c\sqrt{x} + \frac{d}{\ln x} + \frac{e}{\sqrt{x}} \\
5030 \quad y &= a + bx + c\sqrt{x} + \frac{d}{\ln x} + e \frac{\ln x}{x} \\
5031 \quad y &= a + bx + c\sqrt{x} + \frac{d}{\ln x} + \frac{e}{x} \\
5032 \quad y &= a + bx + c\sqrt{x} + \frac{d}{\ln x} + \frac{e}{x\sqrt{x}} \\
5033 \quad y &= a + bx + c\sqrt{x} + \frac{d}{\ln x} + e \frac{\ln x}{x^2} \\
5034 \quad y &= a + bx + c\sqrt{x} + \frac{d}{\ln x} + \frac{e}{x^2}
\end{aligned}$$

$$\begin{aligned}
5035 \quad y &= a + bx + c\sqrt{x} + \frac{d}{\ln x} + e e^{-x} \\
5036 \quad y &= a + bx + c\sqrt{x} + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x} \\
5037 \quad y &= a + bx + c\sqrt{x} + \frac{d}{\sqrt{x}} + \frac{e}{x} \\
5038 \quad y &= a + bx + c\sqrt{x} + \frac{d}{\sqrt{x}} + \frac{e}{x\sqrt{x}} \\
5039 \quad y &= a + bx + c\sqrt{x} + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x^2} \\
5040 \quad y &= a + bx + c\sqrt{x} + \frac{d}{\sqrt{x}} + \frac{e}{x^2} \\
5041 \quad y &= a + bx + c\sqrt{x} + \frac{d}{\sqrt{x}} + e e^{-x} \\
5042 \quad y &= a + bx + c\sqrt{x} + d \frac{\ln x}{x} + \frac{e}{x} \\
5043 \quad y &= a + bx + c\sqrt{x} + d \frac{\ln x}{x} + \frac{e}{x\sqrt{x}} \\
5044 \quad y &= a + bx + c\sqrt{x} + d \frac{\ln x}{x} + e \frac{\ln x}{x^2} \\
5045 \quad y &= a + bx + c\sqrt{x} + d \frac{\ln x}{x} + \frac{e}{x^2} \\
5046 \quad y &= a + bx + c\sqrt{x} + d \frac{\ln x}{x} + e e^{-x} \\
5047 \quad y &= a + bx + c\sqrt{x} + \frac{d}{x} + \frac{e}{x\sqrt{x}} \\
5048 \quad y &= a + bx + c\sqrt{x} + \frac{d}{x} + e \frac{\ln x}{x^2} \\
5049 \quad y &= a + bx + c\sqrt{x} + \frac{d}{x} + \frac{e}{x^2} \\
5050 \quad y &= a + bx + c\sqrt{x} + \frac{d}{x} + e e^{-x} \\
5051 \quad y &= a + bx + c\sqrt{x} + \frac{d}{x\sqrt{x}} + e \frac{\ln x}{x^2} \\
5052 \quad y &= a + bx + c\sqrt{x} + \frac{d}{x\sqrt{x}} + \frac{e}{x^2} \\
5053 \quad y &= a + bx + c\sqrt{x} + \frac{d}{x\sqrt{x}} + e e^{-x} \\
5054 \quad y &= a + bx + c\sqrt{x} + d \frac{\ln x}{x^2} + \frac{e}{x^2} \\
5055 \quad y &= a + bx + c\sqrt{x} + d \frac{\ln x}{x^2} + e e^{-x} \\
5056 \quad y &= a + bx + c\sqrt{x} + \frac{d}{x^2} + e e^{-x} \\
5057 \quad y &= a + bx + c \ln x + \frac{d}{\ln x} + \frac{e}{\sqrt{x}} \\
5058 \quad y &= a + bx + c \ln x + \frac{d}{\ln x} + e \frac{\ln x}{x} \\
5059 \quad y &= a + bx + c \ln x + \frac{d}{\ln x} + \frac{e}{x} \\
5060 \quad y &= a + bx + c \ln x + \frac{d}{\ln x} + \frac{e}{x\sqrt{x}}
\end{aligned}$$

$$\begin{aligned}
5061 \quad y &= a + bx + c \ln x + \frac{d}{\ln x} + e \frac{\ln x}{x^2} \\
5062 \quad y &= a + bx + c \ln x + \frac{d}{\ln x} + \frac{e}{x^2} \\
5063 \quad y &= a + bx + c \ln x + \frac{d}{\ln x} + e e^{-x} \\
5064 \quad y &= a + bx + c \ln x + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x} \\
5065 \quad y &= a + bx + c \ln x + \frac{d}{\sqrt{x}} + \frac{e}{x} \\
5066 \quad y &= a + bx + c \ln x + \frac{d}{\sqrt{x}} + \frac{e}{x\sqrt{x}} \\
5067 \quad y &= a + bx + c \ln x + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x^2} \\
5068 \quad y &= a + bx + c \ln x + \frac{d}{\sqrt{x}} + \frac{e}{x^2} \\
5069 \quad y &= a + bx + c \ln x + \frac{d}{\sqrt{x}} + e e^{-x} \\
5070 \quad y &= a + bx + c \ln x + d \frac{\ln x}{x} + \frac{e}{x} \\
5071 \quad y &= a + bx + c \ln x + d \frac{\ln x}{x} + \frac{e}{x\sqrt{x}} \\
5072 \quad y &= a + bx + c \ln x + d \frac{\ln x}{x} + e \frac{\ln x}{x^2} \\
5073 \quad y &= a + bx + c \ln x + d \frac{\ln x}{x} + \frac{e}{x^2} \\
5074 \quad y &= a + bx + c \ln x + d \frac{\ln x}{x} + e e^{-x} \\
5075 \quad y &= a + bx + c \ln x + \frac{d}{x} + \frac{e}{x\sqrt{x}} \\
5076 \quad y &= a + bx + c \ln x + \frac{d}{x} + e \frac{\ln x}{x^2} \\
5077 \quad y &= a + bx + c \ln x + \frac{d}{x} + \frac{e}{x^2} \\
5078 \quad y &= a + bx + c \ln x + \frac{d}{x} + e e^{-x} \\
5079 \quad y &= a + bx + c \ln x + \frac{d}{x\sqrt{x}} + e \frac{\ln x}{x^2} \\
5080 \quad y &= a + bx + c \ln x + \frac{d}{x\sqrt{x}} + \frac{e}{x^2} \\
5081 \quad y &= a + bx + c \ln x + \frac{d}{x\sqrt{x}} + e e^{-x} \\
5082 \quad y &= a + bx + c \ln x + d \frac{\ln x}{x^2} + \frac{e}{x^2} \\
5083 \quad y &= a + bx + c \ln x + d \frac{\ln x}{x^2} + e e^{-x} \\
5084 \quad y &= a + bx + c \ln x + \frac{d}{x^2} + e e^{-x} \\
5085 \quad y &= a + bx + \frac{c}{\ln x} + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x} \\
5086 \quad y &= a + bx + \frac{c}{\ln x} + \frac{d}{\sqrt{x}} + \frac{e}{x}
\end{aligned}$$

$$5087 \ y = a + bx + \frac{c}{\ln x} + \frac{d}{\sqrt{x}} + \frac{e}{x\sqrt{x}}$$

$$5088 \ y = a + bx + \frac{c}{\ln x} + \frac{d}{\sqrt{x}} + e \frac{\ln x}{x^2}$$

$$5089 \ y = a + bx + \frac{c}{\ln x} + \frac{d}{\sqrt{x}} + \frac{e}{x^2}$$

$$5090 \ y = a + bx + \frac{c}{\ln x} + \frac{d}{\sqrt{x}} + e e^{-x}$$

$$5091 \ y = a + bx + \frac{c}{\ln x} + d \frac{\ln x}{x} + \frac{e}{x}$$

$$5092 \ y = a + bx + \frac{c}{\ln x} + d \frac{\ln x}{x} + \frac{e}{x\sqrt{x}}$$

$$5093 \ y = a + bx + \frac{c}{\ln x} + d \frac{\ln x}{x} + e \frac{\ln x}{x^2}$$

$$5094 \ y = a + bx + \frac{c}{\ln x} + d \frac{\ln x}{x} + \frac{e}{x^2}$$

$$5095 \ y = a + bx + \frac{c}{\ln x} + d \frac{\ln x}{x} + e e^{-x}$$

$$5096 \ y = a + bx + \frac{c}{\ln x} + \frac{d}{x} + \frac{e}{x\sqrt{x}}$$

$$5097 \ y = a + bx + \frac{c}{\ln x} + \frac{d}{x} + e \frac{\ln x}{x^2}$$

$$5098 \ y = a + bx + \frac{c}{\ln x} + \frac{d}{x} + \frac{e}{x^2}$$

$$5099 \ y = a + bx + \frac{c}{\ln x} + \frac{d}{x} + e e^{-x}$$

$$5100 \ y = a + bx + \frac{c}{\ln x} + \frac{d}{x\sqrt{x}} + e \frac{\ln x}{x^2}$$

$$5101 \ y = a + bx + \frac{c}{\ln x} + \frac{d}{x\sqrt{x}} + \frac{e}{x^2}$$

$$5102 \ y = a + bx + \frac{c}{\ln x} + \frac{d}{x\sqrt{x}} + e e^{-x}$$

$$5103 \ y = a + bx + \frac{c}{\ln x} + d \frac{\ln x}{x^2} + \frac{e}{x^2}$$

$$5104 \ y = a + bx + \frac{c}{\ln x} + d \frac{\ln x}{x^2} + e e^{-x}$$

$$5105 \ y = a + bx + \frac{c}{\ln x} + \frac{d}{x^2} + e e^{-x}$$

$$5106 \ y = a + bx + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x} + \frac{e}{x}$$

$$5107 \ y = a + bx + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x} + \frac{e}{x\sqrt{x}}$$

$$5108 \ y = a + bx + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x} + e \frac{\ln x}{x^2}$$

$$5109 \ y = a + bx + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x} + \frac{e}{x^2}$$

$$5110 \ y = a + bx + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x} + e e^{-x}$$

$$5111 \ y = a + bx + \frac{c}{\sqrt{x}} + \frac{d}{x} + \frac{e}{x\sqrt{x}}$$

$$5112 \ y = a + bx + \frac{c}{\sqrt{x}} + \frac{d}{x} + e \frac{\ln x}{x^2}$$

$$5113 \ y = a + bx + \frac{c}{\sqrt{x}} + \frac{d}{x} + \frac{e}{x^2}$$

$$5114 \ y = a + bx + \frac{c}{\sqrt{x}} + \frac{d}{x} + e e^{-x}$$

$$5115 \ y = a + bx + \frac{c}{\sqrt{x}} + \frac{d}{x\sqrt{x}} + e \frac{\ln x}{x^2}$$

$$5116 \ y = a + bx + \frac{c}{\sqrt{x}} + \frac{d}{x\sqrt{x}} + \frac{e}{x^2}$$

$$5117 \ y = a + bx + \frac{c}{\sqrt{x}} + \frac{d}{x\sqrt{x}} + e e^{-x}$$

$$5118 \ y = a + bx + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x^2} + \frac{e}{x^2}$$

$$5119 \ y = a + bx + \frac{c}{\sqrt{x}} + d \frac{\ln x}{x^2} + e e^{-x}$$

$$5120 \ y = a + bx + \frac{c}{\sqrt{x}} + \frac{d}{x^2} + e e^{-x}$$

$$5121 \ y = a + bx + c \frac{\ln x}{x} + \frac{d}{x} + \frac{e}{x\sqrt{x}}$$

$$5122 \ y = a + bx + c \frac{\ln x}{x} + \frac{d}{x} + e \frac{\ln x}{x^2}$$

$$5123 \ y = a + bx + c \frac{\ln x}{x} + \frac{d}{x} + \frac{e}{x^2}$$

$$5124 \ y = a + bx + c \frac{\ln x}{x} + \frac{d}{x} + e e^{-x}$$

$$5125 \ y = a + bx + c \frac{\ln x}{x} + \frac{d}{x\sqrt{x}} + e \frac{\ln x}{x^2}$$

$$5126 \ y = a + bx + c \frac{\ln x}{x} + \frac{d}{x\sqrt{x}} + \frac{e}{x^2}$$

$$5127 \ y = a + bx + c \frac{\ln x}{x} + \frac{d}{x\sqrt{x}} + e e^{-x}$$

$$5128 \ y = a + bx + c \frac{\ln x}{x} + d \frac{\ln x}{x^2} + \frac{e}{x^2}$$

$$5129 \ y = a + bx + c \frac{\ln x}{x} + d \frac{\ln x}{x^2} + e e^{-x}$$

$$5130 \ y = a + bx + c \frac{\ln x}{x} + \frac{d}{x^2} + e e^{-x}$$

$$5131 \ y = a + bx + \frac{c}{x} + \frac{d}{x\sqrt{x}} + e \frac{\ln x}{x^2}$$

$$5132 \ y = a + bx + \frac{c}{x} + \frac{d}{x\sqrt{x}} + \frac{e}{x^2}$$

$$5133 \ y = a + bx + \frac{c}{x} + \frac{d}{x\sqrt{x}} + e e^{-x}$$

$$5134 \ y = a + bx + \frac{c}{x} + d \frac{\ln x}{x^2} + \frac{e}{x^2}$$

$$5135 \ y = a + bx + \frac{c}{x} + d \frac{\ln x}{x^2} + e e^{-x}$$

$$5136 \ y = a + bx + \frac{c}{x} + \frac{d}{x^2} + e e^{-x}$$

$$5137 \ y = a + bx + \frac{c}{x\sqrt{x}} + d \frac{\ln x}{x^2} + \frac{e}{x^2}$$

$$5138 \ y = a + bx + \frac{c}{x\sqrt{x}} + d \frac{\ln x}{x^2} + e e^{-x}$$

$$5139 \ y = a + bx + \frac{c}{x\sqrt{x}} + \frac{d}{x^2} + e e^{-x}$$

$$5140 \ y = a + bx + c \frac{\ln x}{x^2} + \frac{d}{x^2} + e e^{-x}$$

Eqns 6001-6870: Polynomial Equations

Standard Polynomials

- 1 $y = a + bx$
 1003 $y = a + bx + cx^2$
 2040 $y = a + bx + cx^2 + dx^3$
 6001 $y = a + bx + cx^2 + dx^3 + ex^4$
 6002 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5$
 6003 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6$
 6004 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7$
 6005 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8$
 6006 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9$
 6007 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10}$

High-Precision Standard Polynomials

- 6054 $y = a + bx + cx^2 + dx^3 + ex^4$
 6055 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5$
 6056 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6$
 6057 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7$
 6058 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8$
 6059 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9$
 6060 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10}$
 6061 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10} + lx^{11}$
 6062 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10} + lx^{11} + mx^{12}$
 6063 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10} + lx^{11} + mx^{12} + nx^{13}$
 6064 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10} + lx^{11} + mx^{12} + nx^{13} + ox^{14}$
 6065 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10} + lx^{11} + mx^{12} + nx^{13} + ox^{14} + px^{15}$
 6066 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10} + lx^{11} + mx^{12} + nx^{13} + ox^{14} + px^{15} + qx^{16}$
 6067 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10} + lx^{11} + mx^{12} + nx^{13} + ox^{14} + px^{15} + qx^{16} + rx^{17}$
 6068 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10} + lx^{11} + mx^{12} + nx^{13} + ox^{14} + px^{15} + qx^{16} + rx^{17} + sx^{18}$
 6069 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10} + lx^{11} + mx^{12} + nx^{13} + ox^{14} + px^{15} + qx^{16} + rx^{17} + sx^{18} + tx^{19}$
 6070 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10} + lx^{11} + mx^{12} + nx^{13} + ox^{14} + px^{15} + qx^{16} + rx^{17} + sx^{18} + tx^{19} + ux^{20}$

Y-Transformed Polynomials

22	$\ln y = a + bx$	43	$1/y = a + bx$
1213	$\ln y = a + bx + cx^2$	1423	$1/y = a + bx + cx^2$
6101	$\ln y = a + bx + cx^2 + dx^3$	6111	$1/y = a + bx + cx^2 + dx^3$
6102	$\ln y = a + bx + cx^2 + dx^3 + ex^4$	6112	$1/y = a + bx + cx^2 + dx^3 + ex^4$
6103	$\ln y = a + bx + cx^2 + dx^3 + ex^4 + fx^5$	6113	$1/y = a + bx + cx^2 + dx^3 + ex^4 + fx^5$
64	$\sqrt{y} = a + bx$	85	$y^2 = a + bx$
6121	$\sqrt{y} = a + bx + cx^2$	6131	$y^2 = a + bx + cx^2$
6122	$\sqrt{y} = a + bx + cx^2 + dx^3$	6132	$y^2 = a + bx + cx^2 + dx^3$
6123	$\sqrt{y} = a + bx + cx^2 + dx^3 + ex^4$	6133	$y^2 = a + bx + cx^2 + dx^3 + ex^4$
6124	$\sqrt{y} = a + bx + cx^2 + dx^3 + ex^4 + fx^5$	6134	$y^2 = a + bx + cx^2 + dx^3 + ex^4 + fx^5$

Balanced Order Polynomials

1	$y = a + bx$
1016	$y = a + bx + \frac{c}{x}$
2050	$y = a + bx + cx^2 + \frac{d}{x}$
4453	$y = a + bx + cx^2 + \frac{d}{x} + \frac{e}{x^2}$
6201	$y = a + bx + \frac{c}{x} + dx^2 + \frac{e}{x^2} + fx^3$
6202	$y = a + bx + \frac{c}{x} + dx^2 + \frac{e}{x^2} + fx^3 + \frac{g}{x^3}$
6203	$y = a + bx + \frac{c}{x} + dx^2 + \frac{e}{x^2} + fx^3 + \frac{g}{x^3} + hx^4$
6204	$y = a + bx + \frac{c}{x} + dx^2 + \frac{e}{x^2} + fx^3 + \frac{g}{x^3} + hx^4 + \frac{i}{x^4}$
6205	$y = a + bx + \frac{c}{x} + dx^2 + \frac{e}{x^2} + fx^3 + \frac{g}{x^3} + hx^4 + \frac{i}{x^4} + jx^5$
6206	$y = a + bx + \frac{c}{x} + dx^2 + \frac{e}{x^2} + fx^3 + \frac{g}{x^3} + hx^4 + \frac{i}{x^4} + jx^5 + \frac{k}{x^5}$

Natural Logarithm of X Polynomials

$$13 \quad y = a + b \ln x$$

$$1147 \quad y = a + b(\ln x)^2 + c \ln x$$

$$6301 \quad y = a + b \ln x + c(\ln x)^2 + d(\ln x)^3$$

$$6302 \quad y = a + b \ln x + c(\ln x)^2 + d(\ln x)^3 + e(\ln x)^4$$

$$6303 \quad y = a + b \ln x + c(\ln x)^2 + d(\ln x)^3 + e(\ln x)^4 + f(\ln x)^5$$

$$6304 \quad y = a + b \ln x + c(\ln x)^2 + d(\ln x)^3 + e(\ln x)^4 + f(\ln x)^5 + g(\ln x)^6$$

$$6305 \quad y = a + b \ln x + c(\ln x)^2 + d(\ln x)^3 + e(\ln x)^4 + f(\ln x)^5 + g(\ln x)^6 + h(\ln x)^7$$

$$6306 \quad y = a + b \ln x + c(\ln x)^2 + d(\ln x)^3 + e(\ln x)^4 + f(\ln x)^5 + g(\ln x)^6 + h(\ln x)^7 + i(\ln x)^8$$

$$6307 \quad y = a + b \ln x + c(\ln x)^2 + d(\ln x)^3 + e(\ln x)^4 + f(\ln x)^5 + g(\ln x)^6 + h(\ln x)^7 + i(\ln x)^8 + j(\ln x)^9$$

$$6308 \quad y = a + b \ln x + c(\ln x)^2 + d(\ln x)^3 + e(\ln x)^4 + f(\ln x)^5 + g(\ln x)^6 + h(\ln x)^7 + i(\ln x)^8 + j(\ln x)^9 + k(\ln x)^{10}$$

Balanced Natural Logarithm of X Polynomials

$$13 \quad y = a + b \ln x$$

$$1175 \quad y = a + b \ln x + \frac{c}{\ln x}$$

$$3130 \quad y = a + b(\ln x)^2 + c \ln x + \frac{d}{\ln x}$$

$$6401 \quad y = a + b \ln x + \frac{c}{\ln x} + d(\ln x)^2 + \frac{e}{(\ln x)^2}$$

$$6402 \quad y = a + b \ln x + \frac{c}{\ln x} + d(\ln x)^2 + \frac{e}{(\ln x)^2} + f(\ln x)^3$$

$$6403 \quad y = a + b \ln x + \frac{c}{\ln x} + d(\ln x)^2 + \frac{e}{(\ln x)^2} + f(\ln x)^3 + \frac{g}{(\ln x)^3}$$

$$6404 \quad y = a + b \ln x + \frac{c}{\ln x} + d(\ln x)^2 + \frac{e}{(\ln x)^2} + f(\ln x)^3 + \frac{g}{(\ln x)^3} + h(\ln x)^4$$

$$6405 \quad y = a + b \ln x + \frac{c}{\ln x} + d(\ln x)^2 + \frac{e}{(\ln x)^2} + f(\ln x)^3 + \frac{g}{(\ln x)^3} + h(\ln x)^4 + \frac{i}{(\ln x)^4}$$

$$6406 \quad y = a + b \ln x + \frac{c}{\ln x} + d(\ln x)^2 + \frac{e}{(\ln x)^2} + f(\ln x)^3 + \frac{g}{(\ln x)^3} + h(\ln x)^4 + \frac{i}{(\ln x)^4} + j(\ln x)^5$$

$$6407 \quad y = a + b \ln x + \frac{c}{\ln x} + d(\ln x)^2 + \frac{e}{(\ln x)^2} + f(\ln x)^3 + \frac{g}{(\ln x)^3} + h(\ln x)^4 + \frac{i}{(\ln x)^4} + j(\ln x)^5 + \frac{k}{(\ln x)^5}$$

Inverse X Polynomials

$$17 \quad y = a + \frac{b}{x}$$

$$1203 \quad y = a + \frac{b}{x} + \frac{c}{x^2}$$

$$6501 \quad y = a + \frac{b}{x} + \frac{c}{x^2} + \frac{d}{x^3}$$

$$6502 \quad y = a + \frac{b}{x} + \frac{c}{x^2} + \frac{d}{x^3} + \frac{e}{x^4}$$

$$6503 \quad y = a + \frac{b}{x} + \frac{c}{x^2} + \frac{d}{x^3} + \frac{e}{x^4} + \frac{f}{x^5}$$

$$6504 \quad y = a + \frac{b}{x} + \frac{c}{x^2} + \frac{d}{x^3} + \frac{e}{x^4} + \frac{f}{x^5} + \frac{g}{x^6}$$

$$6505 \quad y = a + \frac{b}{x} + \frac{c}{x^2} + \frac{d}{x^3} + \frac{e}{x^4} + \frac{f}{x^5} + \frac{g}{x^6} + \frac{h}{x^7}$$

$$6506 \quad y = a + \frac{b}{x} + \frac{c}{x^2} + \frac{d}{x^3} + \frac{e}{x^4} + \frac{f}{x^5} + \frac{g}{x^6} + \frac{h}{x^7} + \frac{i}{x^8}$$

$$6507 \quad y = a + \frac{b}{x} + \frac{c}{x^2} + \frac{d}{x^3} + \frac{e}{x^4} + \frac{f}{x^5} + \frac{g}{x^6} + \frac{h}{x^7} + \frac{i}{x^8} + \frac{j}{x^9}$$

$$6508 \quad y = a + \frac{b}{x} + \frac{c}{x^2} + \frac{d}{x^3} + \frac{e}{x^4} + \frac{f}{x^5} + \frac{g}{x^6} + \frac{h}{x^7} + \frac{i}{x^8} + \frac{j}{x^9} + \frac{k}{x^{10}}$$

Inverse Natural Logarithm of X Polynomials

$$14 \quad y = a + \frac{b}{\ln x}$$

$$6601 \quad y = a + \frac{b}{\ln x} + \frac{c}{(\ln x)^2}$$

$$6602 \quad y = a + \frac{b}{\ln x} + \frac{c}{(\ln x)^2} + \frac{d}{(\ln x)^3}$$

$$6603 \quad y = a + \frac{b}{\ln x} + \frac{c}{(\ln x)^2} + \frac{d}{(\ln x)^3} + \frac{e}{(\ln x)^4}$$

$$6604 \quad y = a + \frac{b}{\ln x} + \frac{c}{(\ln x)^2} + \frac{d}{(\ln x)^3} + \frac{e}{(\ln x)^4} + \frac{f}{(\ln x)^5}$$

$$6605 \quad y = a + \frac{b}{\ln x} + \frac{c}{(\ln x)^2} + \frac{d}{(\ln x)^3} + \frac{e}{(\ln x)^4} + \frac{f}{(\ln x)^5} + \frac{g}{(\ln x)^6}$$

$$6606 \quad y = a + \frac{b}{\ln x} + \frac{c}{(\ln x)^2} + \frac{d}{(\ln x)^3} + \frac{e}{(\ln x)^4} + \frac{f}{(\ln x)^5} + \frac{g}{(\ln x)^6} + \frac{h}{(\ln x)^7}$$

$$6607 \quad y = a + \frac{b}{\ln x} + \frac{c}{(\ln x)^2} + \frac{d}{(\ln x)^3} + \frac{e}{(\ln x)^4} + \frac{f}{(\ln x)^5} + \frac{g}{(\ln x)^6} + \frac{h}{(\ln x)^7} + \frac{i}{(\ln x)^8}$$

$$6608 \quad y = a + \frac{b}{\ln x} + \frac{c}{(\ln x)^2} + \frac{d}{(\ln x)^3} + \frac{e}{(\ln x)^4} + \frac{f}{(\ln x)^5} + \frac{g}{(\ln x)^6} + \frac{h}{(\ln x)^7} + \frac{i}{(\ln x)^8} + \frac{j}{(\ln x)^9}$$

$$6609 \quad y = a + \frac{b}{\ln x} + \frac{c}{(\ln x)^2} + \frac{d}{(\ln x)^3} + \frac{e}{(\ln x)^4} + \frac{f}{(\ln x)^5} + \frac{g}{(\ln x)^6} + \frac{h}{(\ln x)^7} + \frac{i}{(\ln x)^8} + \frac{j}{(\ln x)^9} + \frac{k}{(\ln x)^{10}}$$

Half Order Polynomials

$$12 \quad y = a + b\sqrt{x}$$

$$1011 \quad y = a + bx + c\sqrt{x}$$

$$2028 \quad y = a + bx + cx\sqrt{x} + d\sqrt{x}$$

$$4179 \quad y = a + bx + cx\sqrt{x} + dx^2 + e\sqrt{x}$$

$$6701 \quad y = a + b\sqrt{x} + cx + dx\sqrt{x} + ex^2 + fx^2\sqrt{x}$$

$$6702 \quad y = a + b\sqrt{x} + cx + dx\sqrt{x} + ex^2 + fx^2\sqrt{x} + gx^3$$

$$6703 \quad y = a + b\sqrt{x} + cx + dx\sqrt{x} + ex^2 + fx^2\sqrt{x} + gx^3 + hx^3\sqrt{x}$$

$$6704 \quad y = a + b\sqrt{x} + cx + dx\sqrt{x} + ex^2 + fx^2\sqrt{x} + gx^3 + hx^3\sqrt{x} + ix^4$$

$$6705 \quad y = a + b\sqrt{x} + cx + dx\sqrt{x} + ex^2 + fx^2\sqrt{x} + gx^3 + hx^3\sqrt{x} + ix^4 + jx^4\sqrt{x}$$

$$6706 \quad y = a + b\sqrt{x} + cx + dx\sqrt{x} + ex^2 + fx^2\sqrt{x} + gx^3 + hx^3\sqrt{x} + ix^4 + jx^4\sqrt{x} + kx^5$$

Even Order Polynomials

$$\begin{aligned}
 4 \quad & y = a + bx^2 \\
 6721 \quad & y = a + bx^2 + cx^4 \\
 6722 \quad & y = a + bx^2 + cx^4 + dx^6 \\
 6723 \quad & y = a + bx^2 + cx^4 + dx^6 + ex^8 \\
 6724 \quad & y = a + bx^2 + cx^4 + dx^6 + ex^8 + fx^{10} \\
 6725 \quad & y = a + bx^2 + cx^4 + dx^6 + ex^8 + fx^{10} + gx^{12} \\
 6726 \quad & y = a + bx^2 + cx^4 + dx^6 + ex^8 + fx^{10} + gx^{12} + hx^{14} \\
 6727 \quad & y = a + bx^2 + cx^4 + dx^6 + ex^8 + fx^{10} + gx^{12} + hx^{14} + ix^{16} \\
 6728 \quad & y = a + bx^2 + cx^4 + dx^6 + ex^8 + fx^{10} + gx^{12} + hx^{14} + ix^{16} + jx^{18} \\
 6729 \quad & y = a + bx^2 + cx^4 + dx^6 + ex^8 + fx^{10} + gx^{12} + hx^{14} + ix^{16} + jx^{18} + kx^{20}
 \end{aligned}$$

Chebyshev Polynomials

$$\begin{aligned}
 6803 \quad & y = a + bT_1(x') + cT_2(x') + dT_3(x') \\
 6804 \quad & y = a + bT_1(x') + cT_2(x') + dT_3(x') + eT_4(x') \\
 6805 \quad & y = a + bT_1(x') + cT_2(x') + dT_3(x') + eT_4(x') + fT_5(x') \\
 6806 \quad & y = a + bT_1(x') + cT_2(x') + dT_3(x') + eT_4(x') + fT_5(x') + gT_6(x') \\
 6807 \quad & y = a + bT_1(x') + cT_2(x') + dT_3(x') + eT_4(x') + fT_5(x') + gT_6(x') + hT_7(x') \\
 6808 \quad & y = a + bT_1(x') + cT_2(x') + dT_3(x') + eT_4(x') + fT_5(x') + gT_6(x') + hT_7(x') + iT_8(x') \\
 6809 \quad & y = a + bT_1(x') + cT_2(x') + dT_3(x') + eT_4(x') + fT_5(x') + gT_6(x') + hT_7(x') + iT_8(x') + jT_9(x') \\
 6810 \quad & y = a + bT_1(x') + cT_2(x') + dT_3(x') + eT_4(x') + fT_5(x') + gT_6(x') + hT_7(x') + iT_8(x') + jT_9(x') + kT_{10}(x') \\
 6811 \quad & y = a + bT_1(x') + cT_2(x') + dT_3(x') + eT_4(x') + fT_5(x') + gT_6(x') + hT_7(x') + iT_8(x') + jT_9(x') + kT_{10}(x') + lT_{11}(x') \\
 6812 \quad & y = a + bT_1(x') + cT_2(x') + dT_3(x') + eT_4(x') + fT_5(x') + gT_6(x') + hT_7(x') + iT_8(x') + jT_9(x') + kT_{10}(x') + lT_{11}(x') + mT_{12}(x') \\
 6813 \quad & y = a + bT_1(x') + cT_2(x') + dT_3(x') + eT_4(x') + fT_5(x') + gT_6(x') + hT_7(x') + iT_8(x') + jT_9(x') + kT_{10}(x') + lT_{11}(x') + mT_{12}(x') \\
 & \quad + nT_{13}(x') \\
 6814 \quad & y = a + bT_1(x') + cT_2(x') + dT_3(x') + eT_4(x') + fT_5(x') + gT_6(x') + hT_7(x') + iT_8(x') + jT_9(x') + kT_{10}(x') + lT_{11}(x') + mT_{12}(x') \\
 & \quad + nT_{13}(x') + oT_{14}(x') \\
 6815 \quad & y = a + bT_1(x') + cT_2(x') + dT_3(x') + eT_4(x') + fT_5(x') + gT_6(x') + hT_7(x') + iT_8(x') + jT_9(x') + kT_{10}(x') + lT_{11}(x') + mT_{12}(x') \\
 & \quad + nT_{13}(x') + oT_{14}(x') + pT_{15}(x') \\
 6816 \quad & y = a + bT_1(x') + cT_2(x') + dT_3(x') + eT_4(x') + fT_5(x') + gT_6(x') + hT_7(x') + iT_8(x') + jT_9(x') + kT_{10}(x') + lT_{11}(x') + mT_{12}(x') \\
 & \quad + nT_{13}(x') + oT_{14}(x') + pT_{15}(x') + qT_{16}(x') \\
 6817 \quad & y = a + bT_1(x') + cT_2(x') + dT_3(x') + eT_4(x') + fT_5(x') + gT_6(x') + hT_7(x') + iT_8(x') + jT_9(x') + kT_{10}(x') + lT_{11}(x') + mT_{12}(x') \\
 & \quad + nT_{13}(x') + oT_{14}(x') + pT_{15}(x') + qT_{16}(x') + rT_{17}(x') \\
 6818 \quad & y = a + bT_1(x') + cT_2(x') + dT_3(x') + eT_4(x') + fT_5(x') + gT_6(x') + hT_7(x') + iT_8(x') + jT_9(x') + kT_{10}(x') + lT_{11}(x') + mT_{12}(x') \\
 & \quad + nT_{13}(x') + oT_{14}(x') + pT_{15}(x') + qT_{16}(x') + rT_{17}(x') + sT_{18}(x') \\
 6819 \quad & y = a + bT_1(x') + cT_2(x') + dT_3(x') + eT_4(x') + fT_5(x') + gT_6(x') + hT_7(x') + iT_8(x') + jT_9(x') + kT_{10}(x') + lT_{11}(x') + mT_{12}(x') \\
 & \quad + nT_{13}(x') + oT_{14}(x') + pT_{15}(x') + qT_{16}(x') + rT_{17}(x') + sT_{18}(x') + tT_{19}(x') \\
 6820 \quad & y = a + bT_1(x') + cT_2(x') + dT_3(x') + eT_4(x') + fT_5(x') + gT_6(x') + hT_7(x') + iT_8(x') + jT_9(x') + kT_{10}(x') + lT_{11}(x') + mT_{12}(x') \\
 & \quad + nT_{13}(x') + oT_{14}(x') + pT_{15}(x') + qT_{16}(x') + rT_{17}(x') + sT_{18}(x') + tT_{19}(x') + uT_{20}(x')
 \end{aligned}$$

Fourier Series Polynomials

- 6841 $y = a + b \cos(x') + c \sin(x')$
- 6842 $y = a + b \cos(x') + c \sin(x') + d \cos(2x') + e \sin(2x')$
- 6843 $y = a + b \cos(x') + c \sin(x') + d \cos(2x') + e \sin(2x') + f \cos(3x') + g \sin(3x')$
- 6844 $y = a + b \cos(x') + c \sin(x') + d \cos(2x') + e \sin(2x') + f \cos(3x') + g \sin(3x') + h \cos(4x') + i \sin(4x')$
- 6845 $y = a + b \cos(x') + c \sin(x') + d \cos(2x') + e \sin(2x') + f \cos(3x') + g \sin(3x') + h \cos(4x') + i \sin(4x') + j \cos(5x') + k \sin(5x')$
- 6846 $y = a + b \cos(x') + c \sin(x') + d \cos(2x') + e \sin(2x') + f \cos(3x') + g \sin(3x') + h \cos(4x') + i \sin(4x') + j \cos(5x') + k \sin(5x') + l \cos(6x') + m \sin(6x')$
- 6847 $y = a + b \cos(x') + c \sin(x') + d \cos(2x') + e \sin(2x') + f \cos(3x') + g \sin(3x') + h \cos(4x') + i \sin(4x') + j \cos(5x') + k \sin(5x') + l \cos(6x') + m \sin(6x') + n \cos(7x') + o \sin(7x')$
- 6848 $y = a + b \cos(x') + c \sin(x') + d \cos(2x') + e \sin(2x') + f \cos(3x') + g \sin(3x') + h \cos(4x') + i \sin(4x') + j \cos(5x') + k \sin(5x') + l \cos(6x') + m \sin(6x') + n \cos(7x') + o \sin(7x') + p \cos(8x') + q \sin(8x')$
- 6849 $y = a + b \cos(x') + c \sin(x') + d \cos(2x') + e \sin(2x') + f \cos(3x') + g \sin(3x') + h \cos(4x') + i \sin(4x') + j \cos(5x') + k \sin(5x') + l \cos(6x') + m \sin(6x') + n \cos(7x') + o \sin(7x') + p \cos(8x') + q \sin(8x') + r \cos(9x') + s \sin(9x')$
- 6850 $y = a + b \cos(x') + c \sin(x') + d \cos(2x') + e \sin(2x') + f \cos(3x') + g \sin(3x') + h \cos(4x') + i \sin(4x') + j \cos(5x') + k \sin(5x') + l \cos(6x') + m \sin(6x') + n \cos(7x') + o \sin(7x') + p \cos(8x') + q \sin(8x') + r \cos(9x') + s \sin(9x') + t \cos(10x') + u \sin(10x')$

Chebyshev Converted Polynomials

- 6853 $y = a + bx + cx^2 + dx^3$
- 6854 $y = a + bx + cx^2 + dx^3 + ex^4$
- 6855 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5$
- 6856 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6$
- 6857 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7$
- 6858 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8$
- 6859 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9$
- 6860 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10}$
- 6861 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10} + lx^{11}$
- 6862 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10} + lx^{11} + mx^{12}$
- 6863 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10} + lx^{11} + mx^{12} + nx^{13}$
- 6864 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10} + lx^{11} + mx^{12} + nx^{13} + ox^{14}$
- 6865 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10} + lx^{11} + mx^{12} + nx^{13} + ox^{14} + px^{15}$
- 6866 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10} + lx^{11} + mx^{12} + nx^{13} + ox^{14} + px^{15} + qx^{16}$
- 6867 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10} + lx^{11} + mx^{12} + nx^{13} + ox^{14} + px^{15} + qx^{16} + rx^{17}$
- 6868 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10} + lx^{11} + mx^{12} + nx^{13} + ox^{14} + px^{15} + qx^{16} + rx^{17} + sx^{18}$
- 6869 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10} + lx^{11} + mx^{12} + nx^{13} + ox^{14} + px^{15} + qx^{16} + rx^{17} + sx^{18} + tx^{19}$
- 6870 $y = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10} + lx^{11} + mx^{12} + nx^{13} + ox^{14} + px^{15} + qx^{16} + rx^{17} + sx^{18} + tx^{19} + ux^{20}$

Eqns 7001-7939: Rational Equations

Standard Rationals

$$7001 \quad y = \frac{a + cx}{1 + bx}$$

$$7002 \quad y = \frac{a + cx}{1 + bx + dx^2}$$

$$7003 \quad y = \frac{a + cx + ex^2}{1 + bx + dx^2}$$

$$7004 \quad y = \frac{a + cx + ex^2}{1 + bx + dx^2 + fx^3}$$

$$7005 \quad y = \frac{a + cx + ex^2 + gx^3}{1 + bx + dx^2 + fx^3}$$

$$7006 \quad y = \frac{a + cx + ex^2 + gx^3}{1 + bx + dx^2 + fx^3 + hx^4}$$

$$7007 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4}{1 + bx + dx^2 + fx^3 + hx^4}$$

$$7008 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5}$$

$$7009 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5}$$

High-Precision Rationals

$$7054 \quad y = \frac{a + cx + ex^2}{1 + bx + dx^2}$$

$$7055 \quad y = \frac{a + cx + ex^2}{1 + bx + dx^2 + fx^3}$$

$$7056 \quad y = \frac{a + cx + ex^2 + gx^3}{1 + bx + dx^2 + fx^3}$$

$$7057 \quad y = \frac{a + cx + ex^2 + gx^3}{1 + bx + dx^2 + fx^3 + hx^4}$$

$$7058 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4}{1 + bx + dx^2 + fx^3 + hx^4}$$

High-Precision Rationals (cont.)

$$7059 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5}$$

$$7060 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5}$$

$$7061 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5 + lx^6}$$

$$7062 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5 + mx^6}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5 + lx^6}$$

$$7063 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5 + mx^6}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5 + lx^6 + nx^7}$$

$$7064 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5 + mx^6 + ox^7}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5 + lx^6 + nx^7}$$

$$7065 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5 + mx^6 + ox^7}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5 + lx^6 + nx^7 + px^8}$$

$$7066 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5 + mx^6 + ox^7 + qx^8}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5 + lx^6 + nx^7 + px^8}$$

$$7067 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5 + mx^6 + ox^7 + qx^8}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5 + lx^6 + nx^7 + px^8 + rx^9}$$

$$7068 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5 + mx^6 + ox^7 + qx^8 + sx^9}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5 + lx^6 + nx^7 + px^8 + rx^9}$$

$$7069 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5 + mx^6 + ox^7 + qx^8 + sx^9}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5 + lx^6 + nx^7 + px^8 + rx^9 + tx^{10}}$$

$$7070 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5 + mx^6 + ox^7 + qx^8 + sx^9 + ux^{10}}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5 + lx^6 + nx^7 + px^8 + rx^9 + tx^{10}}$$

Y-Transformed Rationals

$$7101 \quad \ln y = \frac{a + cx}{1 + bx}$$

$$7113 \quad \sqrt{y} = \frac{a + cx + ex^2}{1 + bx + dx^2}$$

$$7102 \quad \ln y = \frac{a + cx}{1 + bx + dx^2}$$

$$7114 \quad \sqrt{y} = \frac{a + cx + ex^2}{1 + bx + dx^2 + fx^3}$$

$$7103 \quad \ln y = \frac{a + cx + ex^2}{1 + bx + dx^2}$$

$$7121 \quad y^2 = \frac{a + cx}{1 + bx}$$

$$7104 \quad \ln y = \frac{a + cx + ex^2}{1 + bx + dx^2 + fx^3}$$

$$7122 \quad y^2 = \frac{a + cx}{1 + bx + dx^2}$$

$$7111 \quad \sqrt{y} = \frac{a + cx}{1 + bx}$$

$$7123 \quad y^2 = \frac{a + cx + ex^2}{1 + bx + dx^2}$$

$$7112 \quad \sqrt{y} = \frac{a + cx}{1 + bx + dx^2}$$

$$7124 \quad y^2 = \frac{a + cx + ex^2}{1 + bx + dx^2 + fx^3}$$

Natural Logarithm of X Rationals

$$7201 \quad y = \frac{a + c \ln x}{1 + b \ln x}$$

$$7202 \quad y = \frac{a + c \ln x}{1 + b \ln x + d(\ln x)^2}$$

$$7203 \quad y = \frac{a + c \ln x + e(\ln x)^2}{1 + b \ln x + d(\ln x)^2}$$

$$7204 \quad y = \frac{a + c \ln x + e(\ln x)^2}{1 + b \ln x + d(\ln x)^2 + f(\ln x)^3}$$

$$7205 \quad y = \frac{a + c \ln x + e(\ln x)^2 + g(\ln x)^3}{1 + b \ln x + d(\ln x)^2 + f(\ln x)^3}$$

$$7206 \quad y = \frac{a + c \ln x + e(\ln x)^2 + g(\ln x)^3}{1 + b \ln x + d(\ln x)^2 + f(\ln x)^3 + h(\ln x)^4}$$

$$7207 \quad y = \frac{a + c \ln x + e(\ln x)^2 + g(\ln x)^3 + i(\ln x)^4}{1 + b \ln x + d(\ln x)^2 + f(\ln x)^3 + h(\ln x)^4}$$

$$7208 \quad y = \frac{a + c \ln x + e(\ln x)^2 + g(\ln x)^3 + i(\ln x)^4}{1 + b \ln x + d(\ln x)^2 + f(\ln x)^3 + h(\ln x)^4 + j(\ln x)^5}$$

$$7209 \quad y = \frac{a + c \ln x + e(\ln x)^2 + g(\ln x)^3 + i(\ln x)^4 + k(\ln x)^5}{1 + b \ln x + d(\ln x)^2 + f(\ln x)^3 + h(\ln x)^4 + j(\ln x)^5}$$

Square Root Even Order Rationals

$$7301 \quad y^2 = \frac{a + cx^2}{1 + bx^2}$$

$$7302 \quad y^2 = \frac{a + cx^2}{1 + bx^2 + dx^4}$$

$$7303 \quad y^2 = \frac{a + cx^2 + ex^4}{1 + bx^2 + dx^4}$$

$$7304 \quad y^2 = \frac{a + cx^2 + ex^4}{1 + bx^2 + dx^4 + fx^6}$$

$$7305 \quad y^2 = \frac{a + cx^2 + ex^4 + gx^6}{1 + bx^2 + dx^4 + fx^6}$$

$$7306 \quad y^2 = \frac{a + cx^2 + ex^4 + gx^6}{1 + bx^2 + dx^4 + fx^6 + hx^8}$$

$$7307 \quad y^2 = \frac{a + cx^2 + ex^4 + gx^6 + ix^8}{1 + bx^2 + dx^4 + fx^6 + hx^8}$$

$$7308 \quad y^2 = \frac{a + cx^2 + ex^4 + gx^6 + ix^8}{1 + bx^2 + dx^4 + fx^6 + hx^8 + jx^{10}}$$

$$7309 \quad y^2 = \frac{a + cx^2 + ex^4 + gx^6 + ix^8 + kx^{10}}{1 + bx^2 + dx^4 + fx^6 + hx^8 + jx^{10}}$$

Even Order Rationals

$$7401 \quad y = \frac{a + cx^2}{1 + bx^2}$$

$$7402 \quad y = \frac{a + cx^2}{1 + bx^2 + dx^4}$$

$$7403 \quad y = \frac{a + cx^2 + ex^4}{1 + bx^2 + dx^4}$$

$$7404 \quad y = \frac{a + cx^2 + ex^4}{1 + bx^2 + dx^4 + fx^6}$$

$$7405 \quad y = \frac{a + cx^2 + ex^4 + gx^6}{1 + bx^2 + dx^4 + fx^6}$$

$$7406 \quad y = \frac{a + cx^2 + ex^4 + gx^6}{1 + bx^2 + dx^4 + fx^6 + hx^8}$$

$$7407 \quad y = \frac{a + cx^2 + ex^4 + gx^6 + ix^8}{1 + bx^2 + dx^4 + fx^6 + hx^8}$$

$$7408 \quad y = \frac{a + cx^2 + ex^4 + gx^6 + ix^8}{1 + bx^2 + dx^4 + fx^6 + hx^8 + jx^{10}}$$

$$7409 \quad y = \frac{a + cx^2 + ex^4 + gx^6 + ix^8 + kx^{10}}{1 + bx^2 + dx^4 + fx^6 + hx^8 + jx^{10}}$$

Half Order Rationals

$$7501 \quad y = \frac{a + c\sqrt{x}}{1 + b\sqrt{x}}$$

$$7502 \quad y = \frac{a + c\sqrt{x}}{1 + b\sqrt{x} + dx}$$

$$7503 \quad y = \frac{a + c\sqrt{x} + ex}{1 + b\sqrt{x} + dx}$$

$$7504 \quad y = \frac{a + c\sqrt{x} + ex}{1 + b\sqrt{x} + dx + fx\sqrt{x}}$$

$$7505 \quad y = \frac{a + c\sqrt{x} + ex + gx\sqrt{x}}{1 + b\sqrt{x} + dx + fx\sqrt{x}}$$

$$7506 \quad y = \frac{a + c\sqrt{x} + ex + gx\sqrt{x}}{1 + b\sqrt{x} + dx + fx\sqrt{x} + hx^2}$$

$$7507 \quad y = \frac{a + c\sqrt{x} + ex + gx\sqrt{x} + ix^2}{1 + b\sqrt{x} + dx + fx\sqrt{x} + hx^2}$$

$$7508 \quad y = \frac{a + c\sqrt{x} + ex + gx\sqrt{x} + ix^2}{1 + b\sqrt{x} + dx + fx\sqrt{x} + hx^2 + jx^2\sqrt{x}}$$

$$7509 \quad y = \frac{a + c\sqrt{x} + ex + gx\sqrt{x} + ix^2 + kx^2\sqrt{x}}{1 + b\sqrt{x} + dx + fx\sqrt{x} + hx^2 + jx^2\sqrt{x}}$$

Chebyshev Series Rationals

$$7604y = \left[\frac{a + cT_1(x') + eT_2(x')}{1 + bT_1(x') + dT_2(x')} \right]$$

$$7605y = \left[\frac{a + cT_1(x') + eT_2(x')}{1 + bT_1(x') + dT_2(x') + fT_3(x')} \right]$$

$$7606y = \left[\frac{a + cT_1(x') + eT_2(x') + gT_3(x')}{1 + bT_1(x') + dT_2(x') + fT_3(x')} \right]$$

$$7607y = \left[\frac{a + cT_1(x') + eT_2(x') + gT_3(x')}{1 + bT_1(x') + dT_2(x') + fT_3(x') + hT_4(x')} \right]$$

$$7608y = \left[\frac{a + cT_1(x') + eT_2(x') + gT_3(x') + iT_4(x')}{1 + bT_1(x') + dT_2(x') + fT_3(x') + hT_4(x')} \right]$$

$$7609y = \left[\frac{a + cT_1(x') + eT_2(x') + gT_3(x') + iT_4(x')}{1 + bT_1(x') + dT_2(x') + fT_3(x') + hT_4(x') + jT_5(x')} \right]$$

$$7610y = \left[\frac{a + cT_1(x') + eT_2(x') + gT_3(x') + iT_4(x') + kT_5(x')}{1 + bT_1(x') + dT_2(x') + fT_3(x') + hT_4(x') + jT_5(x')} \right]$$

$$7611y = \left[\frac{a + cT_1(x') + eT_2(x') + gT_3(x') + iT_4(x') + kT_5(x')}{1 + bT_1(x') + dT_2(x') + fT_3(x') + hT_4(x') + jT_5(x') + lT_6(x')} \right]$$

$$7612y = \left[\frac{a + cT_1(x') + eT_2(x') + gT_3(x') + iT_4(x') + kT_5(x') + mT_6(x')}{1 + bT_1(x') + dT_2(x') + fT_3(x') + hT_4(x') + jT_5(x') + lT_6(x')} \right]$$

$$7613y = \left[\frac{a + cT_1(x') + eT_2(x') + gT_3(x') + iT_4(x') + kT_5(x') + mT_6(x')}{1 + bT_1(x') + dT_2(x') + fT_3(x') + hT_4(x') + jT_5(x') + lT_6(x') + nT_7(x')} \right]$$

$$7614y = \left[\frac{a + cT_1(x') + eT_2(x') + gT_3(x') + iT_4(x') + kT_5(x') + mT_6(x') + oT_7(x')}{1 + bT_1(x') + dT_2(x') + fT_3(x') + hT_4(x') + jT_5(x') + lT_6(x') + nT_7(x')} \right]$$

$$7615y = \left[\frac{a + cT_1(x') + eT_2(x') + gT_3(x') + iT_4(x') + kT_5(x') + mT_6(x') + oT_7(x')}{1 + bT_1(x') + dT_2(x') + fT_3(x') + hT_4(x') + jT_5(x') + lT_6(x') + nT_7(x') + pT_8(x')} \right]$$

$$7616y = \left[\frac{a + cT_1(x') + eT_2(x') + gT_3(x') + iT_4(x') + kT_5(x') + mT_6(x') + oT_7(x') + qT_8(x')}{1 + bT_1(x') + dT_2(x') + fT_3(x') + hT_4(x') + jT_5(x') + lT_6(x') + nT_7(x') + pT_8(x')} \right]$$

$$7617y = \left[\frac{a + cT_1(x') + eT_2(x') + gT_3(x') + iT_4(x') + kT_5(x') + mT_6(x') + oT_7(x') + qT_8(x')}{1 + bT_1(x') + dT_2(x') + fT_3(x') + hT_4(x') + jT_5(x') + lT_6(x') + nT_7(x') + pT_8(x') + rT_9(x')} \right]$$

$$7618y = \left[\frac{a + cT_1(x') + eT_2(x') + gT_3(x') + iT_4(x') + kT_5(x') + mT_6(x') + oT_7(x') + qT_8(x') + sT_9(x')}{1 + bT_1(x') + dT_2(x') + fT_3(x') + hT_4(x') + jT_5(x') + lT_6(x') + nT_7(x') + pT_8(x') + rT_9(x')} \right]$$

$$7619y = \left[\frac{a + cT_1(x') + eT_2(x') + gT_3(x') + iT_4(x') + kT_5(x') + mT_6(x') + oT_7(x') + qT_8(x') + sT_9(x')}{1 + bT_1(x') + dT_2(x') + fT_3(x') + hT_4(x') + jT_5(x') + lT_6(x') + nT_7(x') + pT_8(x') + rT_9(x') + tT_{10}(x')} \right]$$

$$7620y = \left[\frac{a + cT_1(x') + eT_2(x') + gT_3(x') + iT_4(x') + kT_5(x') + mT_6(x') + oT_7(x') + qT_8(x') + sT_9(x') + uT_{10}(x')}{1 + bT_1(x') + dT_2(x') + fT_3(x') + hT_4(x') + jT_5(x') + lT_6(x') + nT_7(x') + pT_8(x') + rT_9(x') + tT_{10}(x')} \right]$$

Fourier Series Rationals

$$7641 \quad y = \frac{a + c \cos(x') + e \sin(x')}{1 + b \cos(x') + d \sin(x')}$$

$$7642 \quad y = \frac{a + c \cos(x') + e \sin(x') + g \cos(2x') + i \sin(2x')}{1 + b \cos(x') + d \sin(x') + f \cos(2x') + h \sin(2x')}$$

$$7643 \quad y = \frac{a + c \cos(x') + e \sin(x') + g \cos(2x') + i \sin(2x') + k \cos(3x') + m \sin(3x')}{1 + b \cos(x') + d \sin(x') + f \cos(2x') + h \sin(2x') + j \cos(3x') + l \sin(3x')}$$

$$7644 \quad y = \frac{a + c \cos(x') + e \sin(x') + g \cos(2x') + i \sin(2x') + k \cos(3x') + m \sin(3x') + o \cos(4x') + q \sin(4x')}{1 + b \cos(x') + d \sin(x') + f \cos(2x') + h \sin(2x') + j \cos(3x') + l \sin(3x') + n \cos(4x') + p \sin(4x')}$$

$$7645 \quad y = \frac{a + c \cos(x') + e \sin(x') + g \cos(2x') + i \sin(2x') + k \cos(3x') + m \sin(3x') + o \cos(4x') + q \sin(4x') + s \cos(5x') + u \sin(5x')}{1 + b \cos(x') + d \sin(x') + f \cos(2x') + h \sin(2x') + j \cos(3x') + l \sin(3x') + n \cos(4x') + p \sin(4x') + r \cos(5x') + t \sin(5x')}$$

Chebyshev Converted Rationals

$$7654 \quad y = \frac{a + cx + ex^2}{1 + bx + dx^2}$$

$$7655 \quad y = \frac{a + cx + ex^2}{1 + bx + dx^2 + fx^3}$$

$$7656 \quad y = \frac{a + cx + ex^2 + gx^3}{1 + bx + dx^2 + fx^3}$$

$$7657 \quad y = \frac{a + cx + ex^2 + gx^3}{1 + bx + dx^2 + fx^3 + hx^4}$$

$$7658 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4}{1 + bx + dx^2 + fx^3 + hx^4}$$

$$7659 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5}$$

$$7660 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5}$$

$$7661 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5 + lx^6}$$

$$7662 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5 + mx^6}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5 + lx^6}$$

$$7663 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5 + mx^6}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5 + lx^6 + nx^7}$$

$$7664 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5 + mx^6 + ox^7}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5 + lx^6 + nx^7}$$

$$7665 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5 + mx^6 + ox^7}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5 + lx^6 + nx^7 + px^8}$$

$$7666 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5 + mx^6 + ox^7 + qx^8}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5 + lx^6 + nx^7 + px^8}$$

$$7667 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5 + mx^6 + ox^7 + qx^8}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5 + lx^6 + nx^7 + px^8 + rx^9}$$

$$7668 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5 + mx^6 + ox^7 + qx^8 + sx^9}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5 + lx^6 + nx^7 + px^8 + rx^9}$$

$$7669 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5 + mx^6 + ox^7 + qx^8 + sx^9}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5 + lx^6 + nx^7 + px^8 + rx^9 + tx^{10}}$$

$$7670 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5 + mx^6 + ox^7 + qx^8 + sx^9 + ux^{10}}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5 + lx^6 + nx^7 + px^8 + rx^9 + tx^{10}}$$

Constrained Non-Linearly Fit SVD Rationals

$$7901 \quad y = \frac{a + cx}{1 + bx}$$

$$7902 \quad y = \frac{a + cx}{1 + bx + dx^2}$$

$$7903 \quad y = \frac{a + cx + ex^2}{1 + bx + dx^2}$$

$$7904 \quad y = \frac{a + cx + ex^2}{1 + bx + dx^2 + fx^3}$$

$$7905 \quad y = \frac{a + cx + ex^2 + gx^3}{1 + bx + dx^2 + fx^3}$$

$$7911 \quad y = \frac{a + c \ln x}{1 + b \ln x}$$

$$7912 \quad y = \frac{a + c \ln x}{1 + b \ln x + d(\ln x)^2}$$

$$7913 \quad y = \frac{a + c \ln x + e(\ln x)^2}{1 + b \ln x + d(\ln x)^2}$$

$$7914 \quad y = \frac{a + c \ln x + e(\ln x)^2}{1 + b \ln x + d(\ln x)^2 + f(\ln x)^3}$$

$$7915 \quad y = \frac{a + c \ln x + e(\ln x)^2 + g(\ln x)^3}{1 + b \ln x + d(\ln x)^2 + f(\ln x)^3}$$

$$7916 \quad y = \frac{a + c \ln x + e(\ln x)^2 + g(\ln x)^3}{1 + b \ln x + d(\ln x)^2 + f(\ln x)^3 + h(\ln x)^4}$$

$$7917 \quad y = \frac{a + c \ln x + e(\ln x)^2 + g(\ln x)^3 + i(\ln x)^4}{1 + b \ln x + d(\ln x)^2 + f(\ln x)^3 + h(\ln x)^4}$$

$$7918 \quad y = \frac{a + c \ln x + e(\ln x)^2 + g(\ln x)^3 + i(\ln x)^4}{1 + b \ln x + d(\ln x)^2 + f(\ln x)^3 + h(\ln x)^4 + j(\ln x)^5}$$

$$7919 \quad y = \frac{a + c \ln x + e(\ln x)^2 + g(\ln x)^3 + i(\ln x)^4 + k(\ln x)^5}{1 + b \ln x + d(\ln x)^2 + f(\ln x)^3 + h(\ln x)^4 + j(\ln x)^5}$$

$$7906 \quad y = \frac{a + cx + ex^2 + gx^3}{1 + bx + dx^2 + fx^3 + hx^4}$$

$$7907 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4}{1 + bx + dx^2 + fx^3 + hx^4}$$

$$7908 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5}$$

$$7909 \quad y = \frac{a + cx + ex^2 + gx^3 + ix^4 + kx^5}{1 + bx + dx^2 + fx^3 + hx^4 + jx^5}$$

Constrained Non-Linearly Fit SVD Rationals (cont.)

$$7921 \quad y = \frac{a + cx^2}{1 + bx^2}$$

$$7922 \quad y = \frac{a + cx^2}{1 + bx^2 + dx^4}$$

$$7923 \quad y = \frac{a + cx^2 + ex^4}{1 + bx^2 + dx^4}$$

$$7924 \quad y = \frac{a + cx^2 + ex^4}{1 + bx^2 + dx^4 + fx^6}$$

$$7925 \quad y = \frac{a + cx^2 + ex^4 + gx^6}{1 + bx^2 + dx^4 + fx^6}$$

$$7926 \quad y = \frac{a + cx^2 + ex^4 + gx^6}{1 + bx^2 + dx^4 + fx^6 + hx^8}$$

$$7927 \quad y = \frac{a + cx^2 + ex^4 + gx^6 + ix^8}{1 + bx^2 + dx^4 + fx^6 + hx^8}$$

$$7928 \quad y = \frac{a + cx^2 + ex^4 + gx^6 + ix^8}{1 + bx^2 + dx^4 + fx^6 + hx^8 + jx^{10}}$$

$$7929 \quad y = \frac{a + cx^2 + ex^4 + gx^6 + ix^8 + kx^{10}}{1 + bx^2 + dx^4 + fx^6 + hx^8 + jx^{10}}$$

$$7931 \quad y = \frac{a + c\sqrt{x}}{1 + b\sqrt{x}}$$

$$7932 \quad y = \frac{a + c\sqrt{x}}{1 + b\sqrt{x} + dx}$$

$$7933 \quad y = \frac{a + c\sqrt{x} + ex}{1 + b\sqrt{x} + dx}$$

$$7934 \quad y = \frac{a + c\sqrt{x} + ex}{1 + b\sqrt{x} + dx + fx\sqrt{x}}$$

$$7935 \quad y = \frac{a + c\sqrt{x} + ex + gx\sqrt{x}}{1 + b\sqrt{x} + dx + fx\sqrt{x}}$$

$$7936 \quad y = \frac{a + c\sqrt{x} + ex + gx\sqrt{x}}{1 + b\sqrt{x} + dx + fx\sqrt{x} + hx^2}$$

$$7937 \quad y = \frac{a + c\sqrt{x} + ex + gx\sqrt{x} + ix^2}{1 + b\sqrt{x} + dx + fx\sqrt{x} + hx^2}$$

$$7938 \quad y = \frac{a + c\sqrt{x} + ex + gx\sqrt{x} + ix^2}{1 + b\sqrt{x} + dx + fx\sqrt{x} + hx^2 + jx^2\sqrt{x}}$$

$$7939 \quad y = \frac{a + c\sqrt{x} + ex + gx\sqrt{x} + ix^2 + kx^2\sqrt{x}}{1 + b\sqrt{x} + dx + fx\sqrt{x} + hx^2 + jx^2\sqrt{x}}$$

Non-Linear Equations

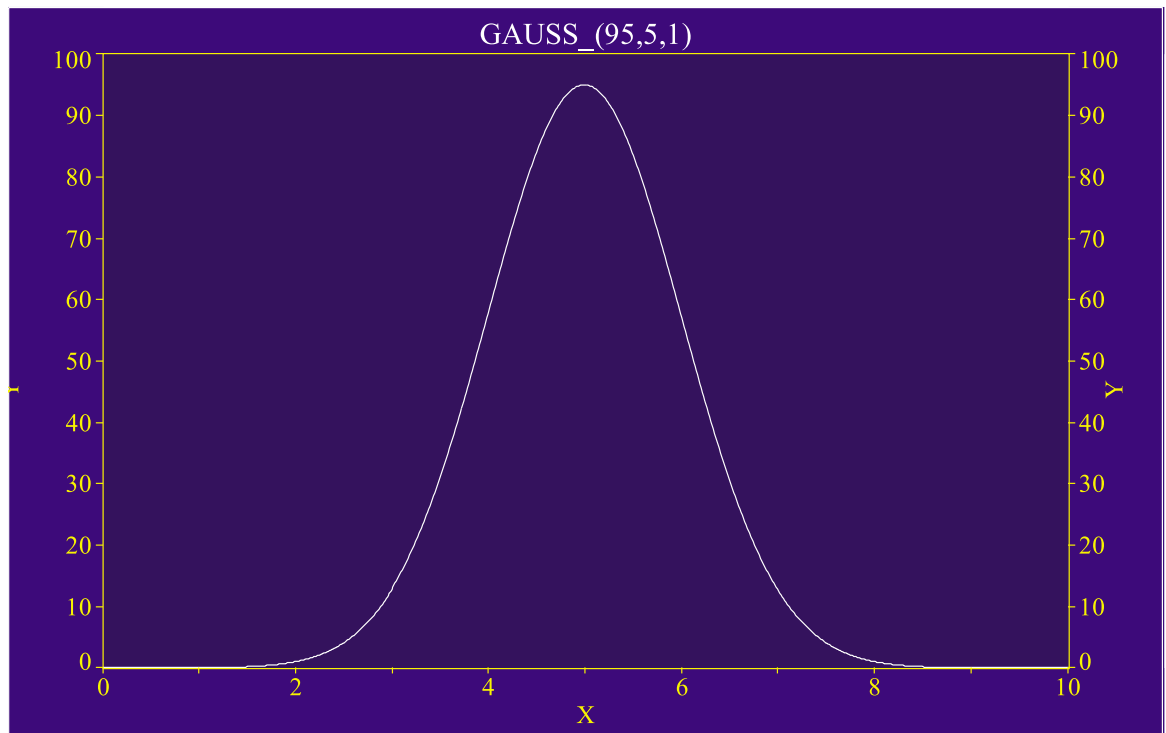
This document furnishes a mathematical description of the various built-in non-linear equations offered by TableCurve 2D.

Symmetric Peak Functions

- Gaussian
- Lorentzian
- Logistic
- Erfc Peak
- Modified Gaussian
- Pearson VII
- Gauss-Lorentz Cross Product
- Symmetric Double Sigmoidal (SDS)
- Symmetric Double Cumulative (SDC)
- Laplace
- Error
- Student's t

These peak functions exhibit a symmetry about the peak center. FWHM is the full width at half-maximum, the width of the peak at 50% of its amplitude.

Gaussian



$$y = a \exp \left[-\frac{1}{2} \left(\frac{x-b}{c} \right)^2 \right]$$

Non-Intercept Form: Equation **8030**

Intercept Form: Equation **8003**

Amplitude: a

Center: b

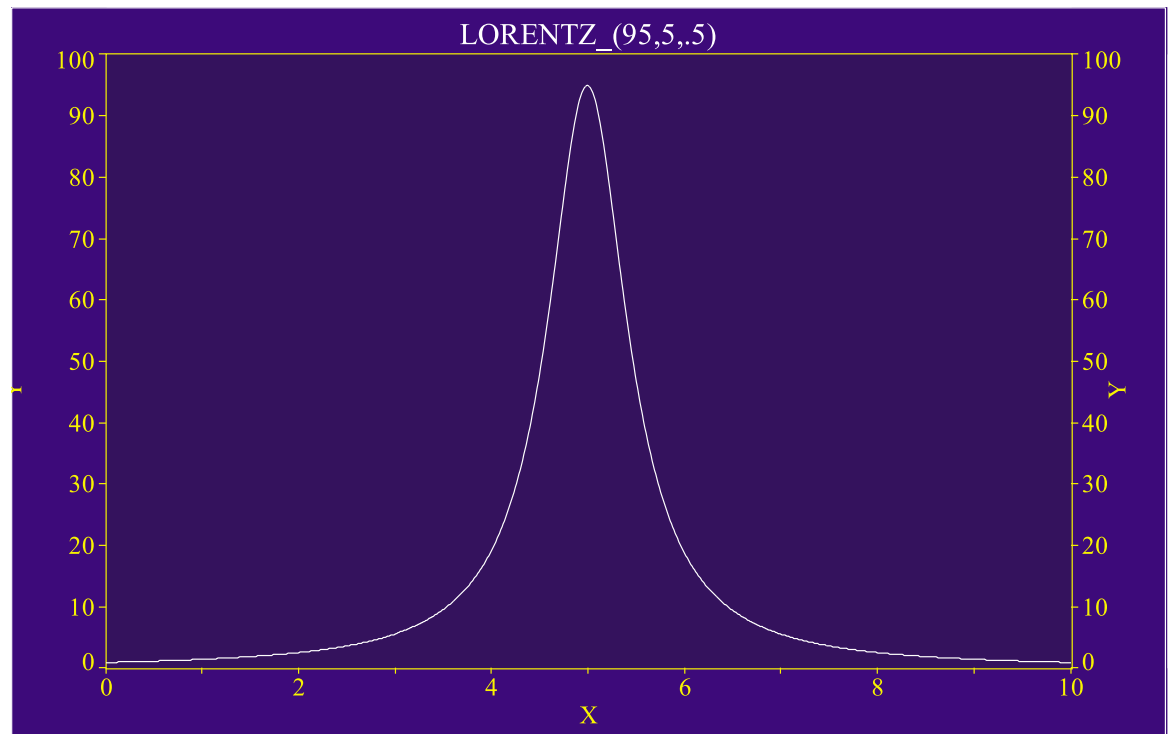
Area: $\sqrt{2\pi}ac$

FWHM: $2\sqrt{2 \ln 2}c=2.354820044c$

Constraints: $c>0$

Built-in Function: **GAUSS_(a,b,c)**

Lorentzian



$$y = \frac{a}{1 + \left(\frac{x-b}{c}\right)^2}$$

Non-Intercept Form: Equation **8031**

Intercept Form: Equation **8004**

Amplitude: a

Center: b

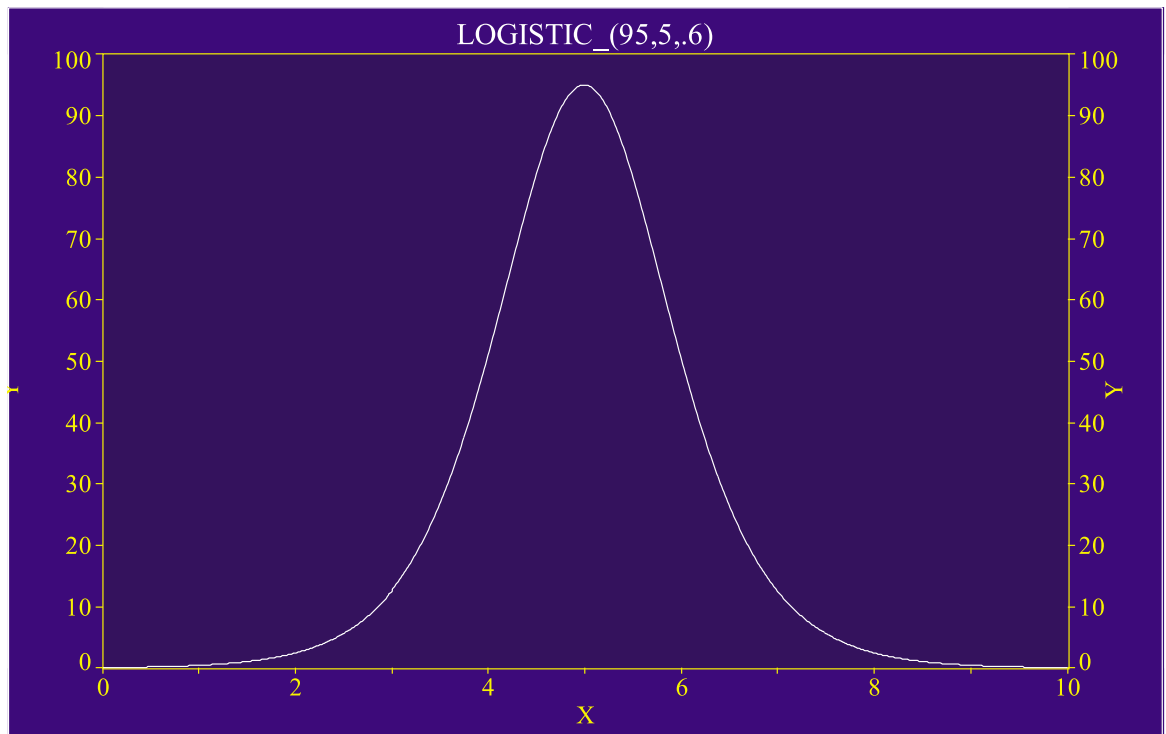
Area: πac

FWHM: $2c$

Constraints: $c > 0$

Built-in Function: **LORENTZ_(a,b,c)**

Logistic Peak



$$y = 4a \frac{\exp\left[-\left(\frac{x-b}{c}\right)\right]}{\left[1 + \exp\left[-\left(\frac{x-b}{c}\right)\right]\right]^2}$$

Non-Intercept Form: Equation **8034**

Intercept Form: Equation **8007**

Amplitude: a

Center: b

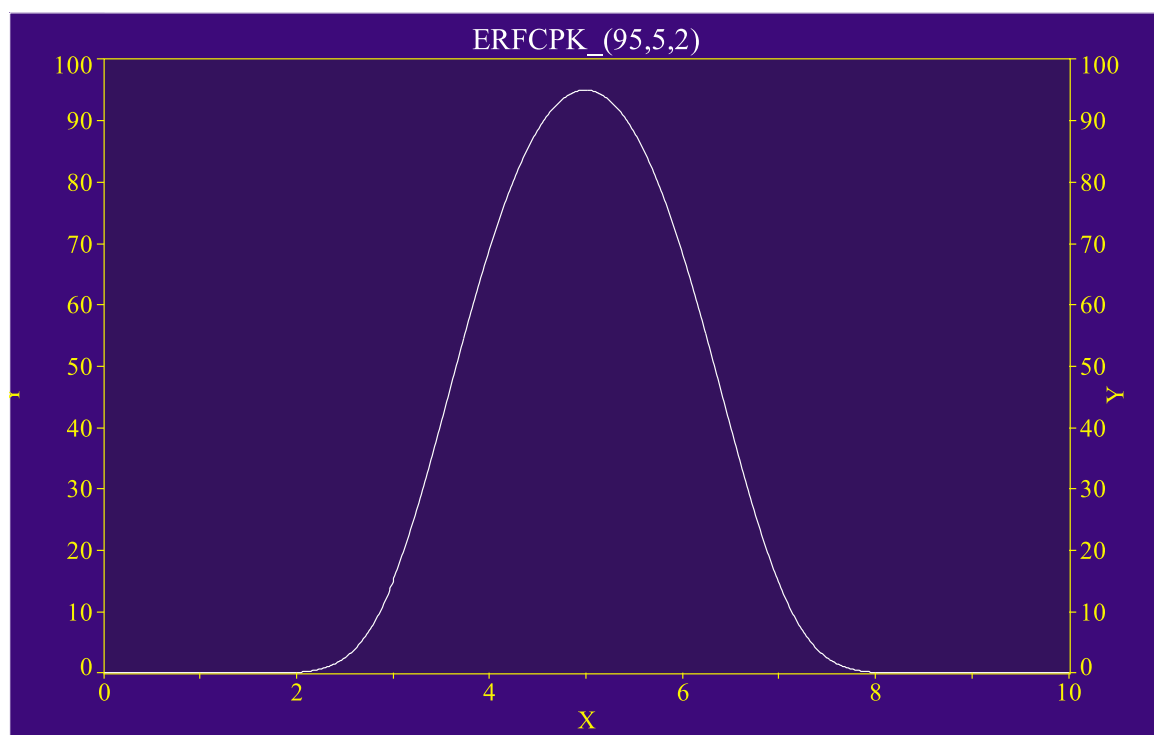
Area: $4ac$

FWHM: $c \ln\left(\frac{3+2\sqrt{2}}{3-2\sqrt{2}}\right) = 3.525494348c$

Constraints: $c > 0$

Built-in Function: **LOGISTIC_(a,b,c)**

Complementary Error Function Peak



$$y = a \operatorname{erfc} \left[\left(\frac{x-b}{c} \right)^2 \right]$$

Non-Intercept Form: Equation **8035**

Intercept Form: Equation **8008**

Amplitude: a

Center: b

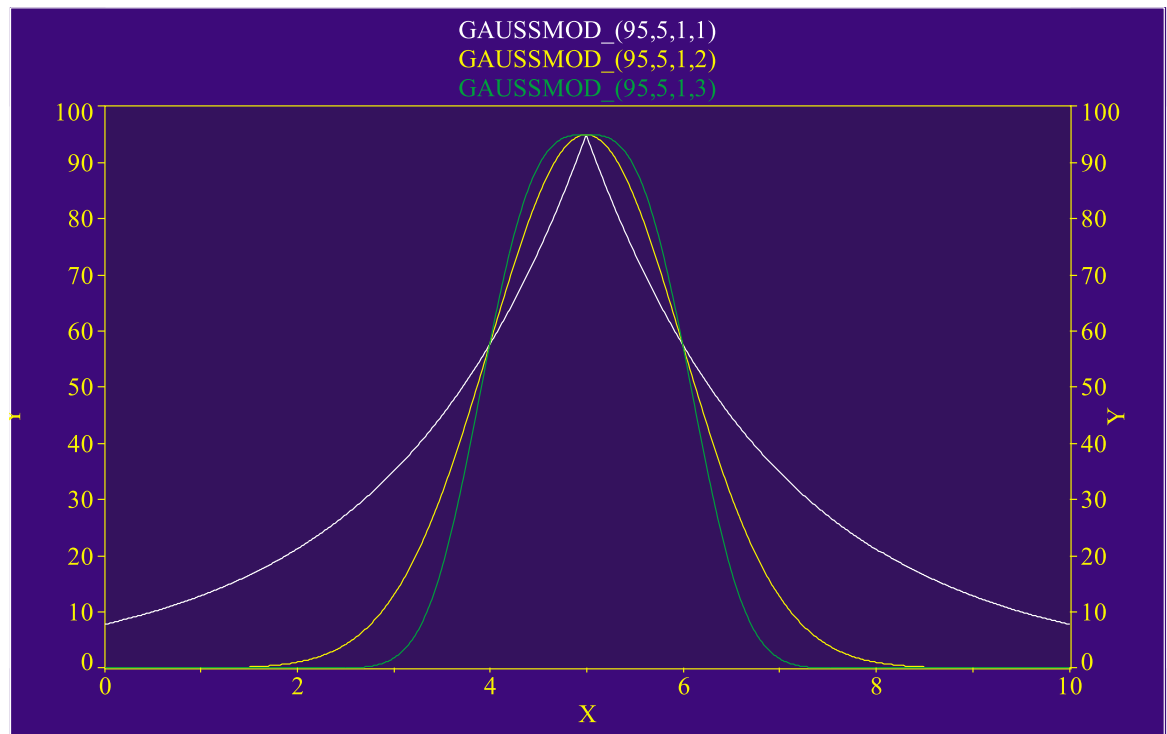
Area: Analytic Solution Unknown

FWHM: $1.381211406c$

Constraints: $c > 0$

Built-in Function: **ERFCPK_**(a,b,c)

Modified Gaussian



$$y = a \exp \left[-\frac{1}{2} \left(\frac{|x - b|}{c} \right)^d \right]$$

Non-Intercept Form: Equation **8038**

Intercept Form: Equation **8039**

Amplitude: a

Center: b

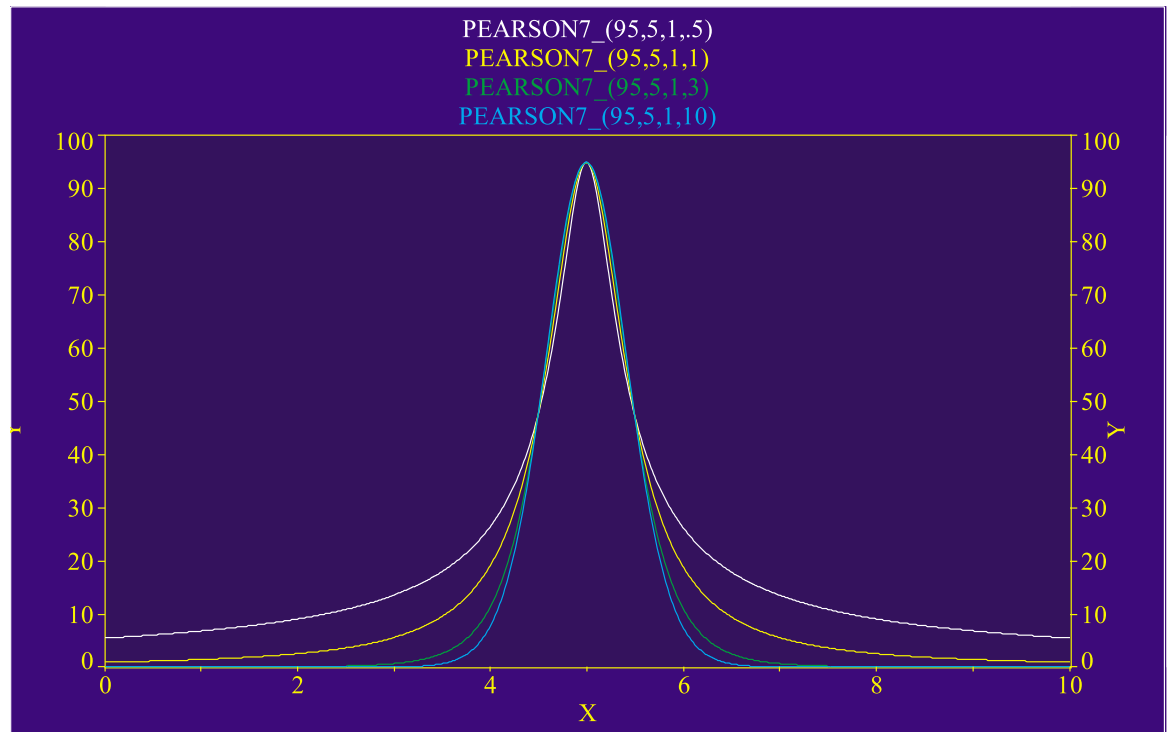
Area: Analytic Solution Unknown

FWHM: $2c(2 \ln 2)^{\frac{1}{d}}$

Constraints: $c > 0, d \geq 1$

Built-in Function: **GAUSSMOD_(a,b,c,d)**

Pearson VII



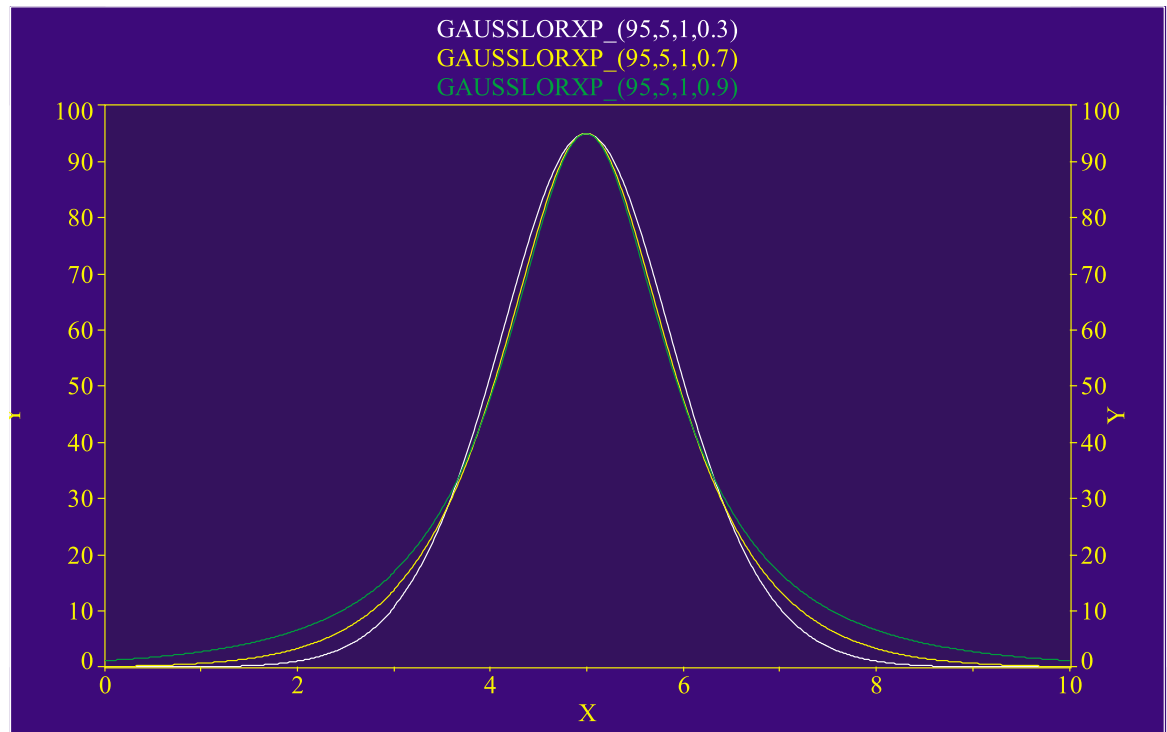
$$y = \frac{a}{\left(1 + \frac{4(x-b)^2 \left(2^{\frac{1}{d}} - 1\right)}{c^2}\right)^d}$$

Non-Intercept Form: Equation **8044**Intercept Form: Equation **8045**Amplitude: a Center: b

$$\text{Area: } \frac{ac\sqrt{\pi d}\Gamma\left(d - \frac{1}{2}\right)}{2d\Gamma(d)\sqrt{\frac{2^{1/d} - 1}{d}}}$$

FWHM: c Constraints: $c > 0, d > 0$ Built-in Function: **PEARSON7_** (a,b,c,d)

Gaussian-Lorentzian Cross Product



$$y = \frac{a}{\left[1 + d\left(\frac{x-b}{c}\right)^2\right] \exp\left[\frac{1-d}{2}\left(\frac{x-b}{c}\right)^2\right]}$$

Non-Intercept Form: Equation **8072**

Intercept Form: Equation **8073**

Amplitude: a

Center: b

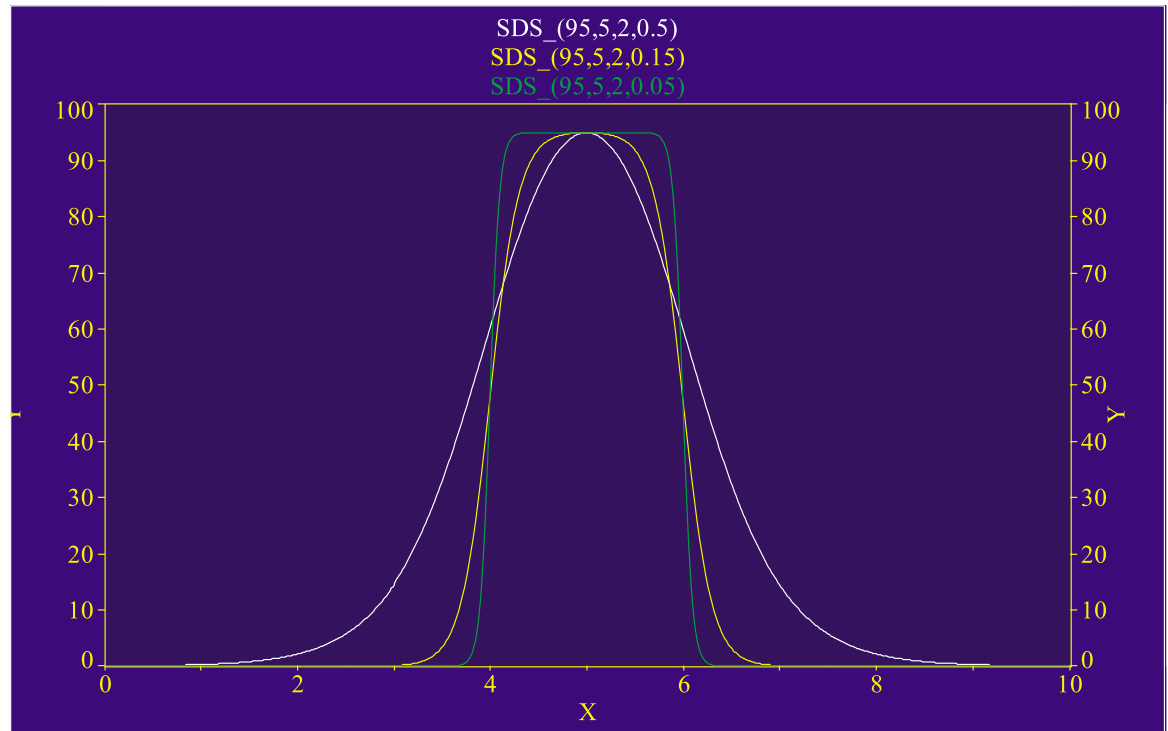
Area: Analytic Solution Unknown

FWHM: Analytic Solution Unknown

Constraints: $c > 0$, $0 \leq d \leq 1$

Built-in Function: **GAUSSLORXP_(a,b,c,d)**

Symmetric Double Sigmoidal (SDS)



$$y = \frac{a \left[1 + \exp\left(\frac{-c}{2d}\right) \right] \left[1 + \exp\left(\frac{c}{2d}\right) \right] \exp\left(-\frac{x-b}{d}\right)}{\left[1 + \exp\left(-\frac{x-b+\frac{c}{2}}{d}\right) \right] \left[1 + \exp\left(-\frac{x-b-\frac{c}{2}}{d}\right) \right]}$$

Non-Intercept Form: Equation **8046**

Intercept Form: Equation **8047**

Amplitude: a

Center: b

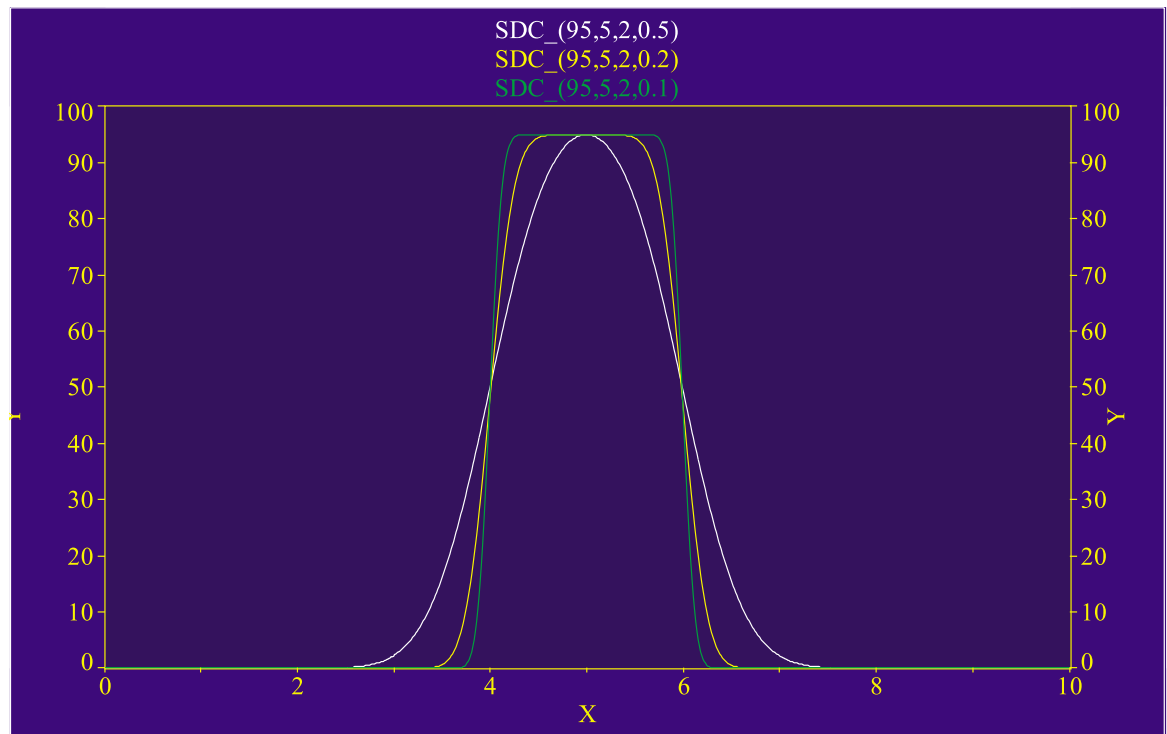
$$\text{Area: } \frac{ac \exp\left(\frac{c}{2d}\right) \left[2 + \exp\left(\frac{-c}{2d}\right) + \exp\left(\frac{c}{2d}\right) \right]}{\exp\left(\frac{c}{d}\right) - 1}$$

$$\text{FWHM: } 2d \ln 2 + c - 2d \ln \left[4 \exp\left(\frac{c}{2d}\right) + \exp\left(\frac{c}{d}\right) + 1 - \sqrt{14 \exp\left(\frac{c}{d}\right) + 8 \exp\left(\frac{3c}{2d}\right) + 8 \exp\left(\frac{c}{2d}\right) + \exp\left(\frac{2c}{d}\right) + 1} \right]$$

Constraints: $c > 0, d > 0$

Built-in Function: **SDS_(a,b,c,d)**

Symmetric Double Gaussian Cumulative (SDC)



$$y = \frac{a \left[1 + \operatorname{erf} \left(\frac{x - b + \frac{c}{2}}{\sqrt{2d}} \right) \right] \left[1 - \operatorname{erf} \left(\frac{x - b - \frac{c}{2}}{\sqrt{2d}} \right) \right]}{\left[1 + \operatorname{erf} \left(\frac{c}{2\sqrt{2d}} \right) \right]^2}$$

Non-Intercept Form: Equation **8048**

Intercept Form: Equation **8049**

Amplitude: a

Center: b

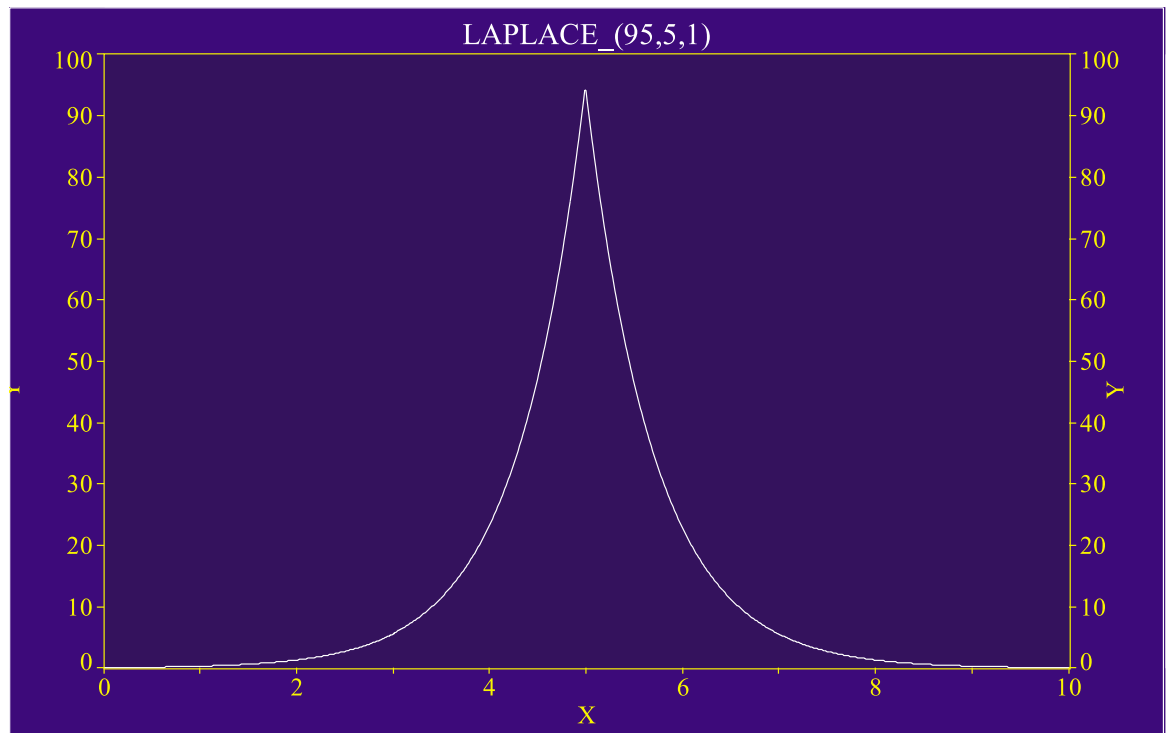
Area: Analytic Solution Unknown

FWHM: Analytic Solution Unknown

Constraints: $c > 0$, $d > 0$

Built-in Function: **SDC_(a,b,c,d)**

Laplace



$$y = a \exp\left(-\frac{\sqrt{2}|x - b|}{c}\right)$$

Non-Intercept Form: Equation **8168**

Intercept Form: Equation **8169**

Amplitude: a

Center: b

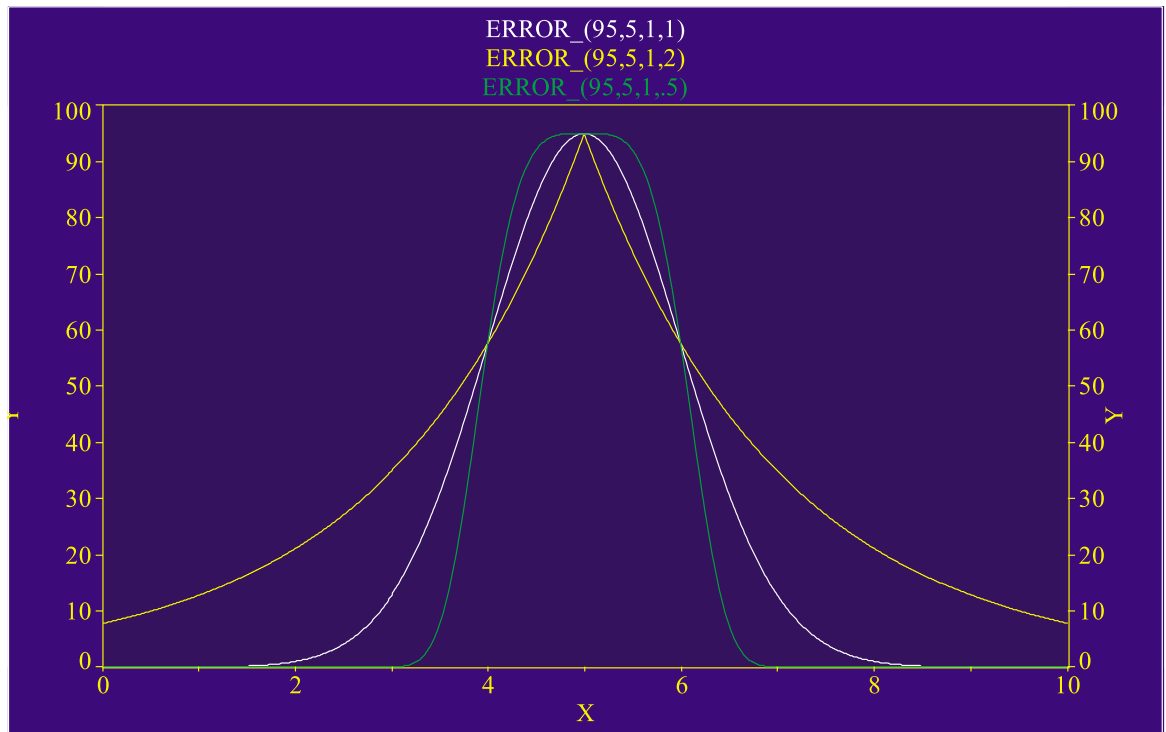
Area: $\sqrt{2}ac$

FWHM: $\sqrt{2}c \ln 2$

Constraints: $c > 0$

Built-in Function: **LAPLACE_(a,b,c)**

Error



$$y = a \exp \left[-\frac{1}{2} \frac{|x-b|^{\frac{2}{d}}}{c} \right]$$

Non-Intercept Form: Equation **8170**

Intercept Form: Equation **8171**

Amplitude: a

Center: b

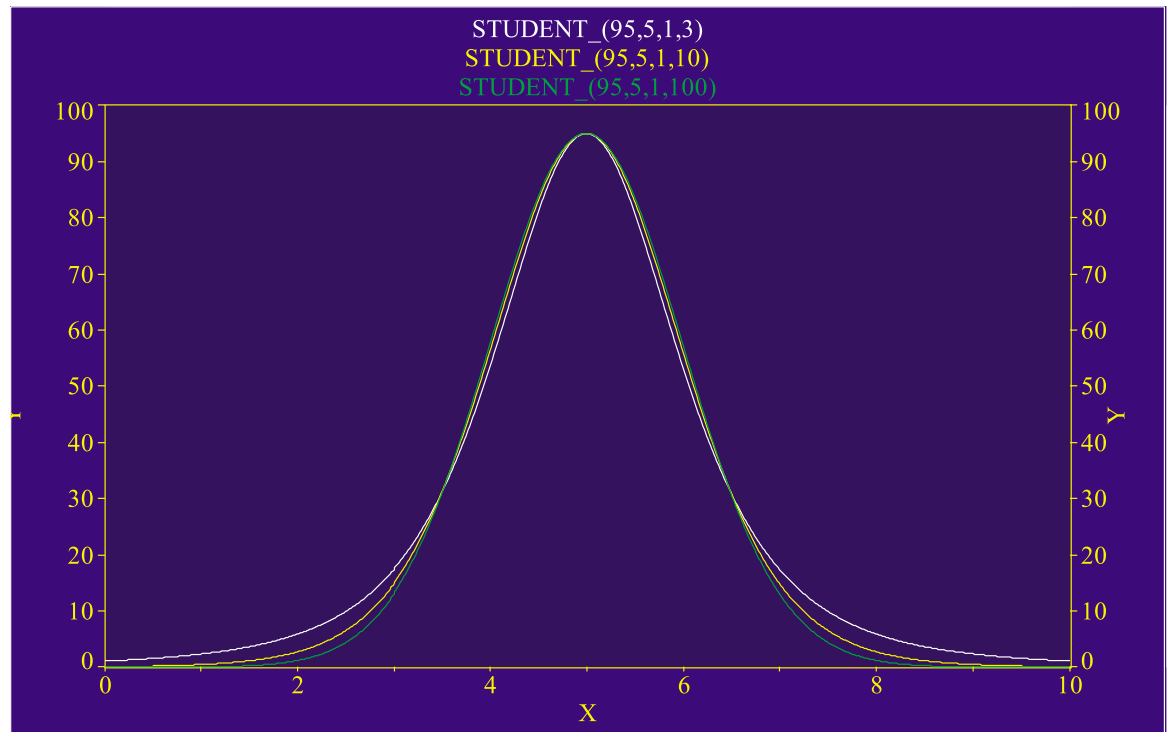
$$\text{Area: } ac^{\frac{d}{2}} 2^{\frac{d}{2}+1} \Gamma\left(\frac{d}{2}+1\right)$$

$$\text{FWHM: } 2(2c \ln 2)^{\frac{d}{2}}$$

Constraints: $c>0, d>0$

Built-in Function: **ERROR_(a,b,c,d)**

Student's t



$$y = \frac{a}{\left(1 + \frac{(x-b)^2}{c^2 d}\right)^{\frac{d+1}{2}}}$$

Non-Intercept Form: Equation **8172**

Intercept Form: Equation **8173**

Amplitude: a

Center: b

$$\text{Area: } \frac{ac\sqrt{\pi d}\Gamma\left(\frac{d}{2}\right)}{\Gamma\left(\frac{d+1}{2}\right)}$$

$$\text{FWHM: } 2cd\sqrt{\frac{4^{\frac{1}{d+1}} - 1}{d}}$$

Constraints: $c > 0, d > 0$

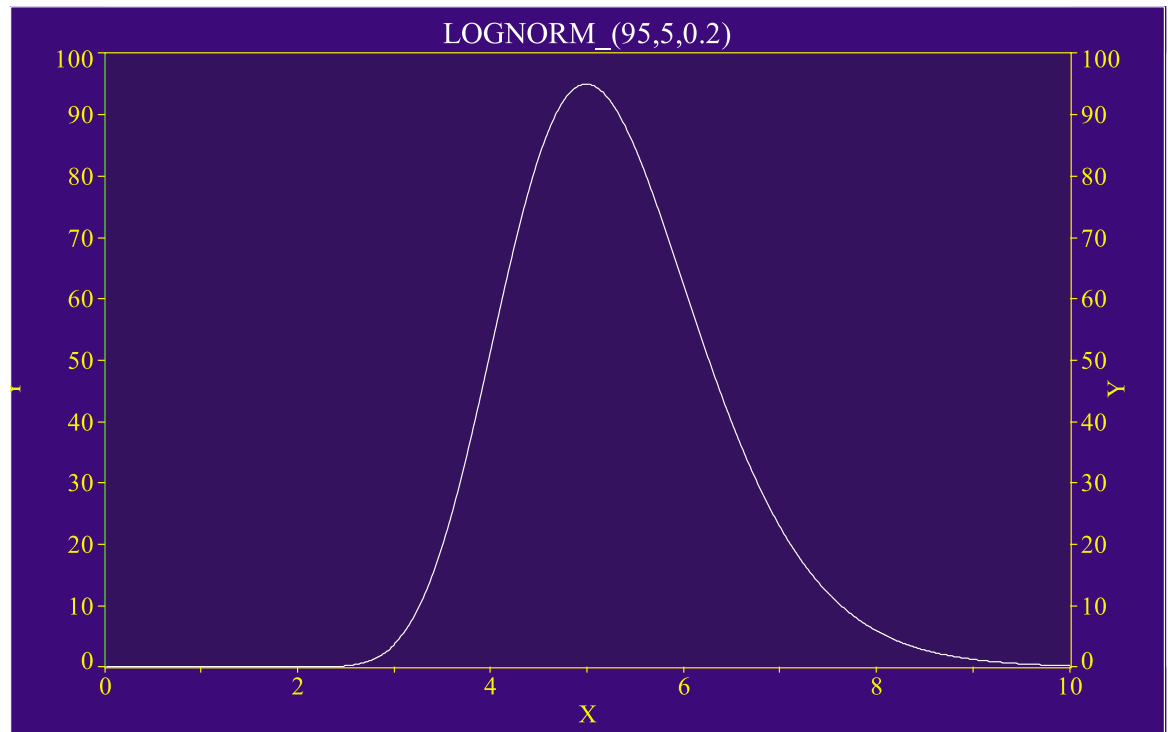
Built-in Function: **STUDENT_(a,b,c,d)**

Asymmetric Peak Functions

- Log Normal
- Extreme Value
- Pulse Peak
- Logistic Dose Response Peak
- Exponential to Max Power Peak
- Asymmetric Logistic Peak
- Gamma
- Weibull
- Logistic Power Peak
- Pulse Peak Modified with Power Term
- Pulse Peak with Second Width Term
- Intermediate Peak
- Beta
- Asymmetric Double Sigmoidal (ADS)
- Asymmetric Double Gaussian Cumulative (ADC)
- Equilibrium Peak
- Exponentially Modified Gaussian (EMG)
- Half-Gaussian Modified Gaussian (GMG)
- Log Normal 4-Parameter
- Extreme Value 4-Parameter Tailed
- Extreme Value 4-Parameter Fronted
- Inverted Gamma
- Chi-Squared
- F-variance
- Pearson IV

These peak functions exhibit an asymmetry about the peak center. FWHM is the full width at half-maximum, the width of the peak at 50% of its amplitude. When a function can model both left and right shifted peaks, both types will be illustrated in the accompanying graph.

Log Normal



$$y = a \exp \left[-\frac{1}{2} \left(\frac{\ln \left(\frac{x}{b} \right)}{c} \right)^2 \right]$$

Non-Intercept Form: Equation **8032**

Intercept Form: Equation **8005**

Amplitude: a

X at Max: b

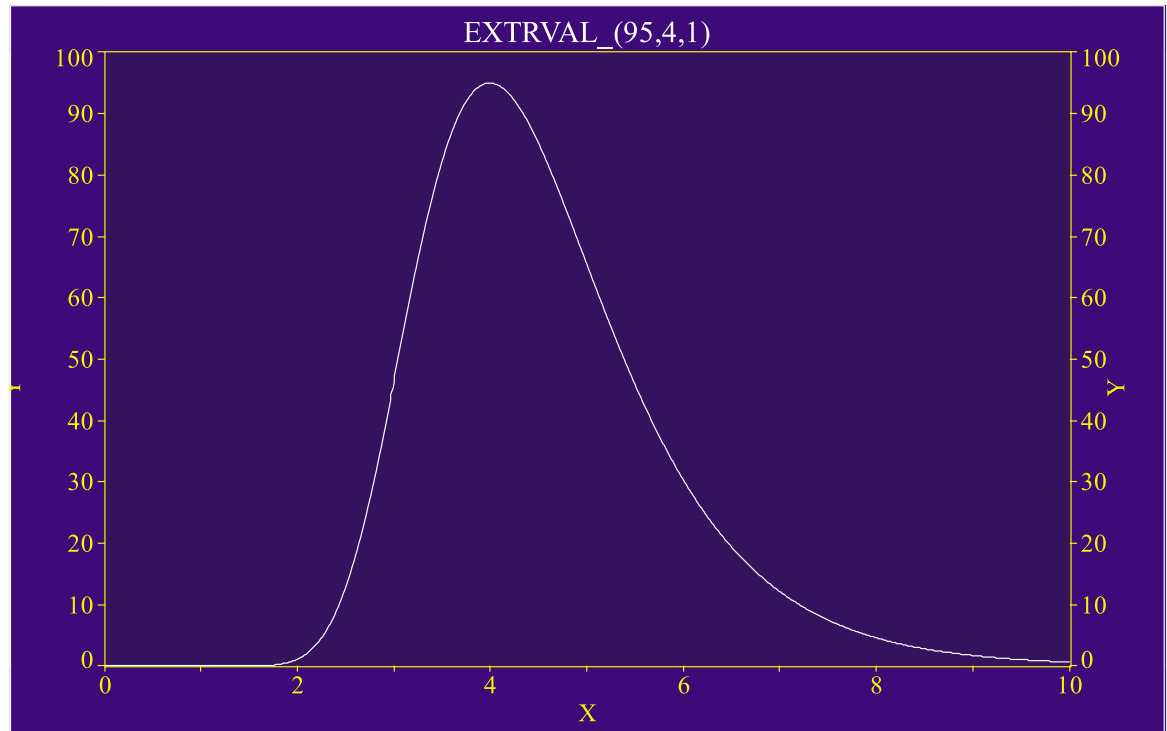
Area: $abc\sqrt{2\pi\exp(c^2)}$

FWHM: $b\exp(c\sqrt{2\ln 2}) - b\exp(-c\sqrt{2\ln 2})$

Constraints: $b \neq 0, c > 0$

Built-in Function: **LOGNORM_(a,b,c)**

Extreme Value



$$y = a \exp \left[-\exp \left(-\frac{x-b}{c} \right) - \frac{x-b}{c} + 1 \right]$$

Non-Intercept Form: Equation **8033**

Intercept Form: Equation **8006**

Amplitude: a

X at Max: b

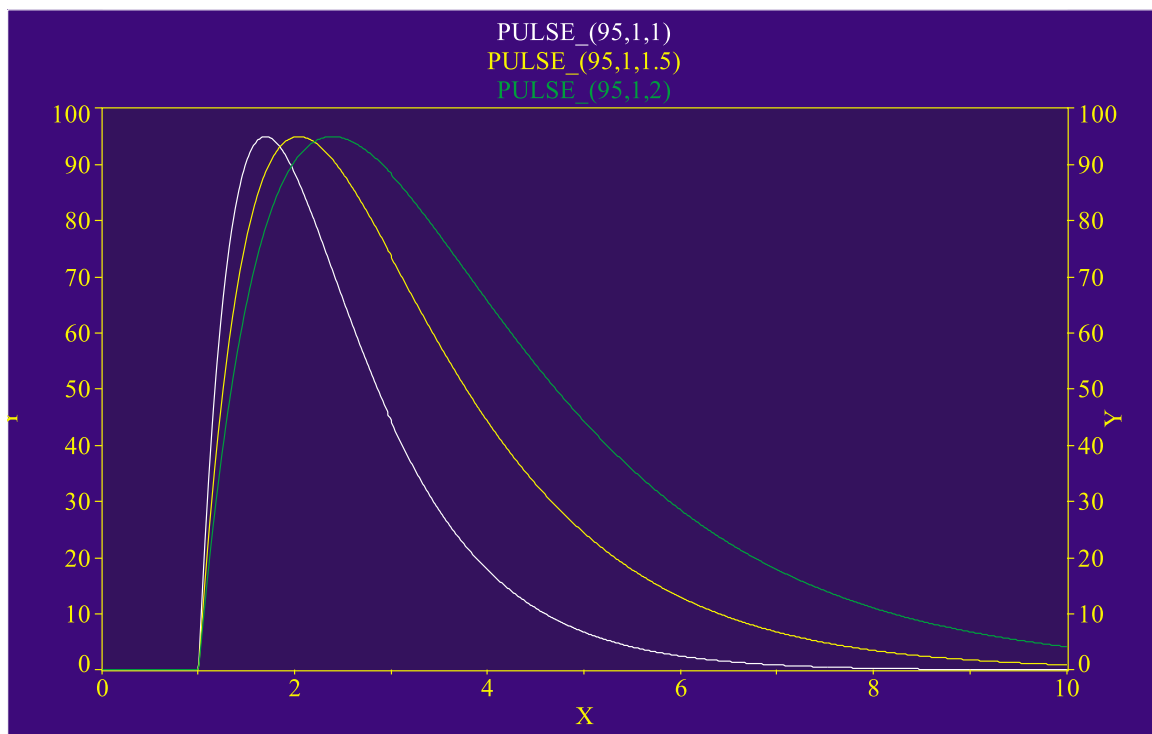
Area: $2.718281828ac$

FWHM: $2.446386037c$

Constraints: $c > 0$

Built-in Function: **EXTRVAL_(a,b,c)**

Pulse Peak



$$y = 4a \exp\left(-\frac{x-b}{c}\right) \left(1 - \exp\left(-\frac{x-b}{c}\right)\right)$$

Non-Intercept Form: Equation **8036**

Intercept Form: Equation **8037**

Amplitude: a

Pulse Initiation: b

X at Max: $b + c \ln 2$

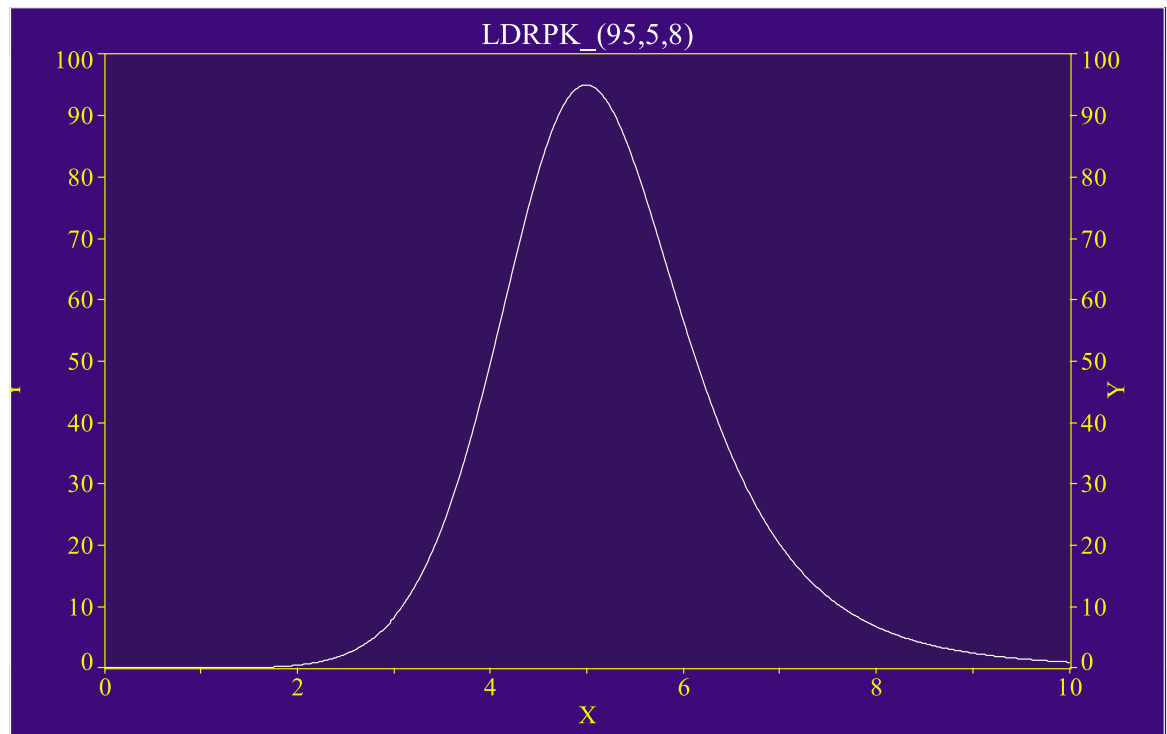
Area: $2ac$

FWHM: $c \left(\ln(2 + \sqrt{2}) - \ln(2 - \sqrt{2}) \right) = 1.762747173c$

Constraints: $x \geq b, c > 0$

Built-in Function: **PULSE_(a, b, c)**

Logistic Dose Response Peak (LDR Peak)



$$y = \frac{4ax^{-c-1}b^{c+1}c^2}{(c-1+cx^{-c}b^c+x^{-c}b^c)^2}$$

Non-Intercept Form: Equation **8040**

Intercept Form: Equation **8041**

Amplitude: a

X at Max: b

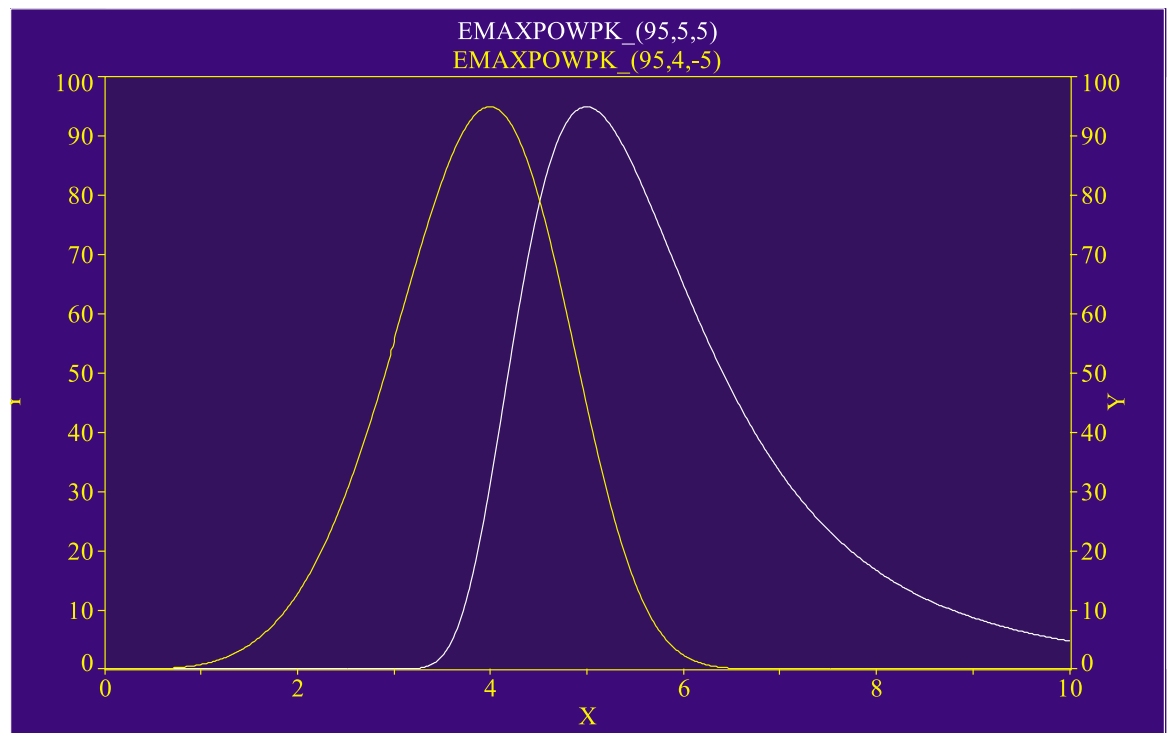
Area: $\frac{4abc}{(c-1)(c+1)}$

FWHM: Analytic Solution Unknown

Constraints: $x > 0$, $b > 0$, $c > 1$

Built-in Function: **LDRPK_(a,b,c)**

Exponential to Max Power Peak



$$y = ax^{-c-1} b^{c+1} \exp\left[\frac{(b^c x^{-c} - 1)(-c - 1)}{c}\right]$$

Non-Intercept Form: Equation **8060**

Intercept Form: Equation **8061**

Amplitude: a

X at Max: b

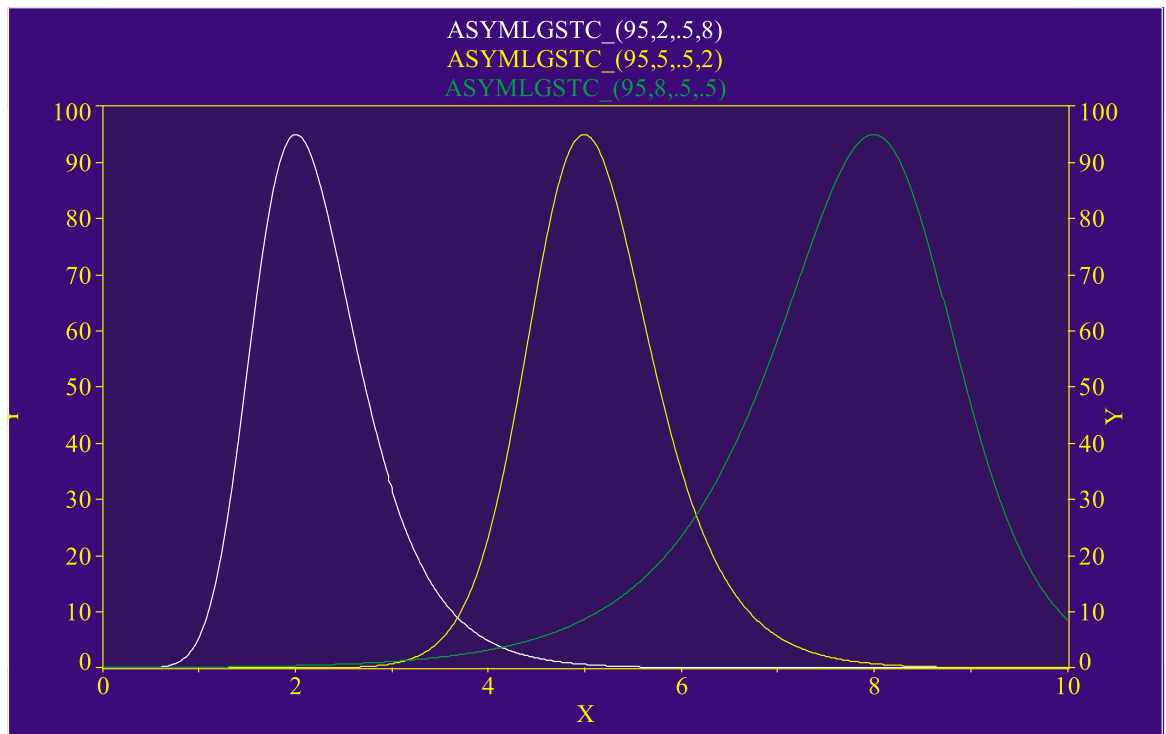
$$\text{Area: } \frac{ab \exp\left(\frac{c+1}{c}\right)}{c+1}$$

FWHM: Analytic Solution Unknown

Constraints: $x > 0$, $b > 0$, $c \geq 2$ or $c \leq -2$

Built-in Function: **EMAXPOWPK_(a,b,c)**

Asymmetric Logistic Peak



$$y = a \left[1 + \exp \left(- \frac{x + c \ln d - b}{c} \right) \right]^{-d-1} d^{-d} (d+1)^{d+1} \exp \left(- \frac{x + c \ln d - b}{c} \right)$$

Non-Intercept Form: Equation **8042**

Intercept Form: Equation **8043**

Amplitude: a

X at Max: b

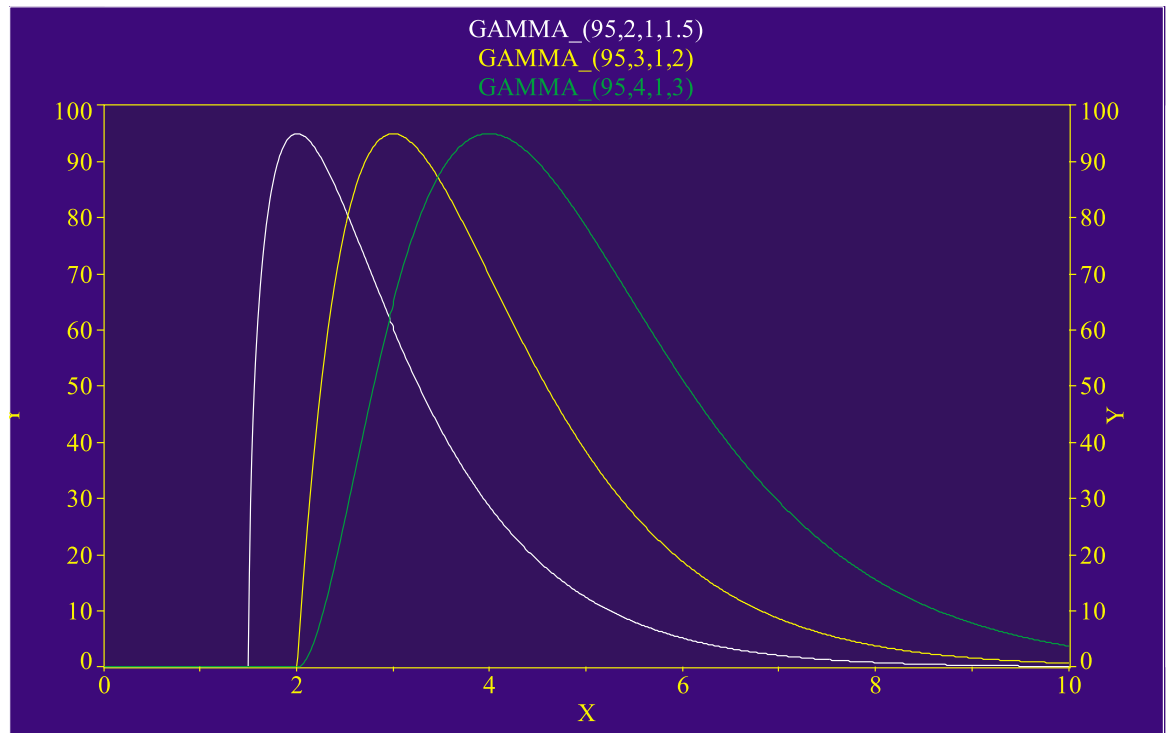
Area: $acd^{-d-1}(d+1)^{d+1}$

FWHM: Analytic Solution Unknown

Constraints: $c > 0, d > 0$

Built-in Function: **ASYMLGSTC_(a,b,c,d)**

Gamma



$$y = a \exp\left(-\frac{x-b}{c}\right) \left(\frac{\frac{x-b}{c} + d - 1}{d - 1}\right)^{d-1}$$

Non-Intercept Form: Equation **8050**

Intercept Form: Equation **8051**

Amplitude: a

X at Max: b

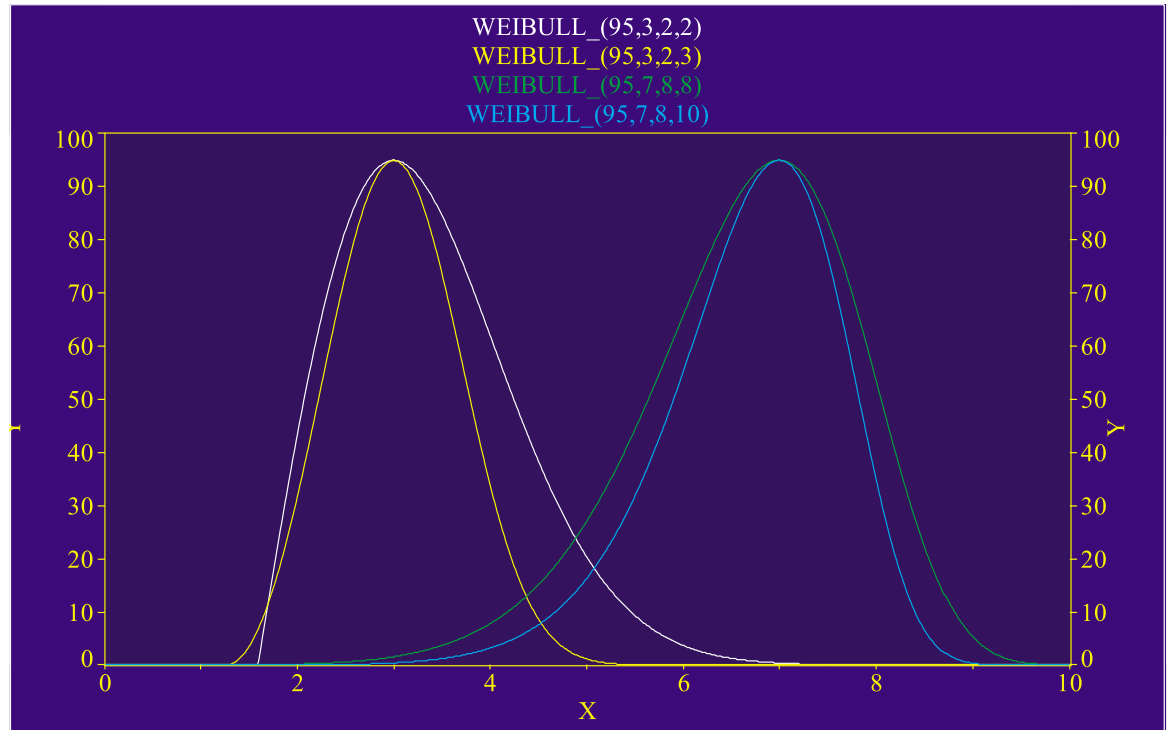
$$\text{Area: } \frac{ac\Gamma(d)}{(d-1)^{d-1} \exp(-d+1)}$$

FWHM: Analytic Solution Unknown

Constraints: $x > (b - cd + c)$, $c > 0$, $d > 1$

Built-in Function: **GAMMA_(a,b,c,d)**

Weibull



$$y = a \left(\frac{d-1}{d} \right)^{\frac{1-d}{d}} \left[\frac{x-b}{c} + \left(\frac{d-1}{d} \right)^{\frac{1}{d}} \right]^{d-1} \exp \left[- \left(\frac{x-b}{c} + \left(\frac{d-1}{d} \right)^{\frac{1}{d}} \right)^d + \frac{d-1}{d} \right]$$

Non-Intercept Form: Equation **8052**

Intercept Form: Equation **8053**

Amplitude: a

X at Max: b

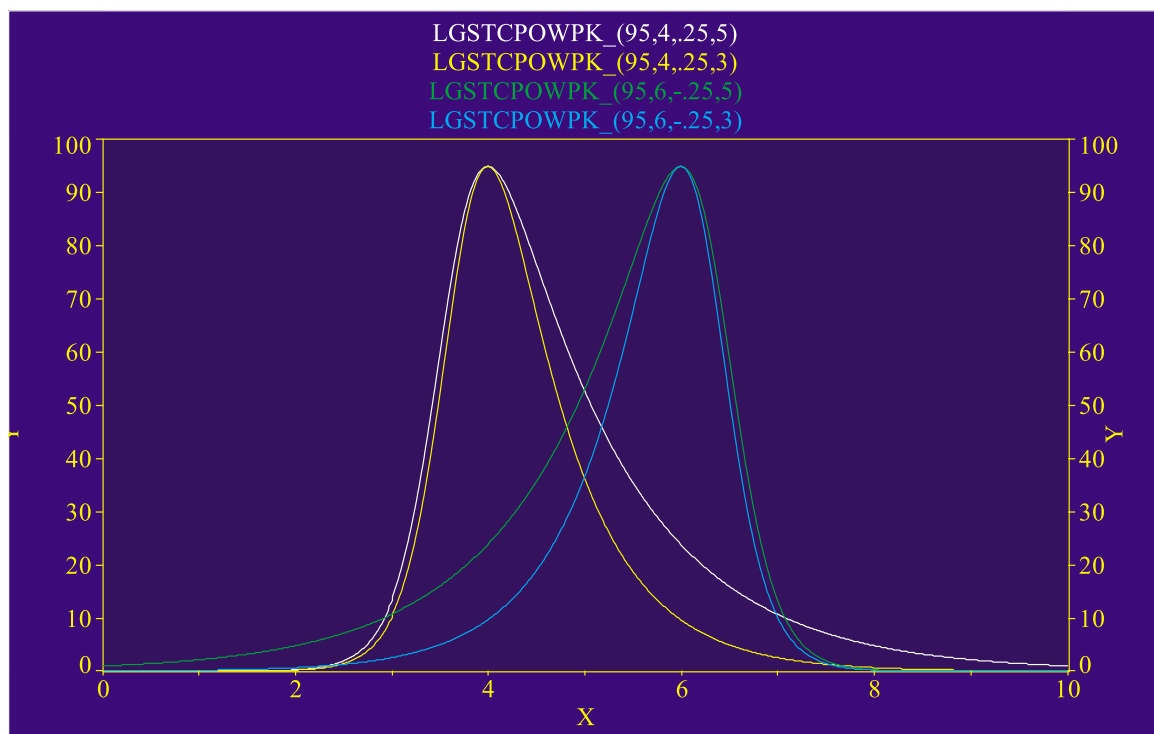
$$\text{Area: } \frac{ac^d}{d} \left[c \left(\frac{d-1}{d} \right)^{\frac{1}{d}} \right]^{-d+1} \exp \left(\frac{d-1}{d} \right)$$

FWHM: Analytic Solution Unknown

Constraints: $x > b - c \left(\frac{d-1}{d} \right)^{\frac{1}{d}}$, $c > 0$, $d > 1$

Built-in Function: **WEIBULL**_ (a,b,c,d)

Logistic Power Peak



$$y = \frac{a}{d} \left[1 + \exp\left(\frac{x + c \ln d - b}{c}\right) \right]^{-\frac{d+1}{d}} \exp\left(\frac{x + c \ln d - b}{c}\right) (d+1)^{\frac{d+1}{d}}$$

Non-Intercept Form: Equation **8062**

Intercept Form: Equation **8063**

Amplitude: a

X at Max: b

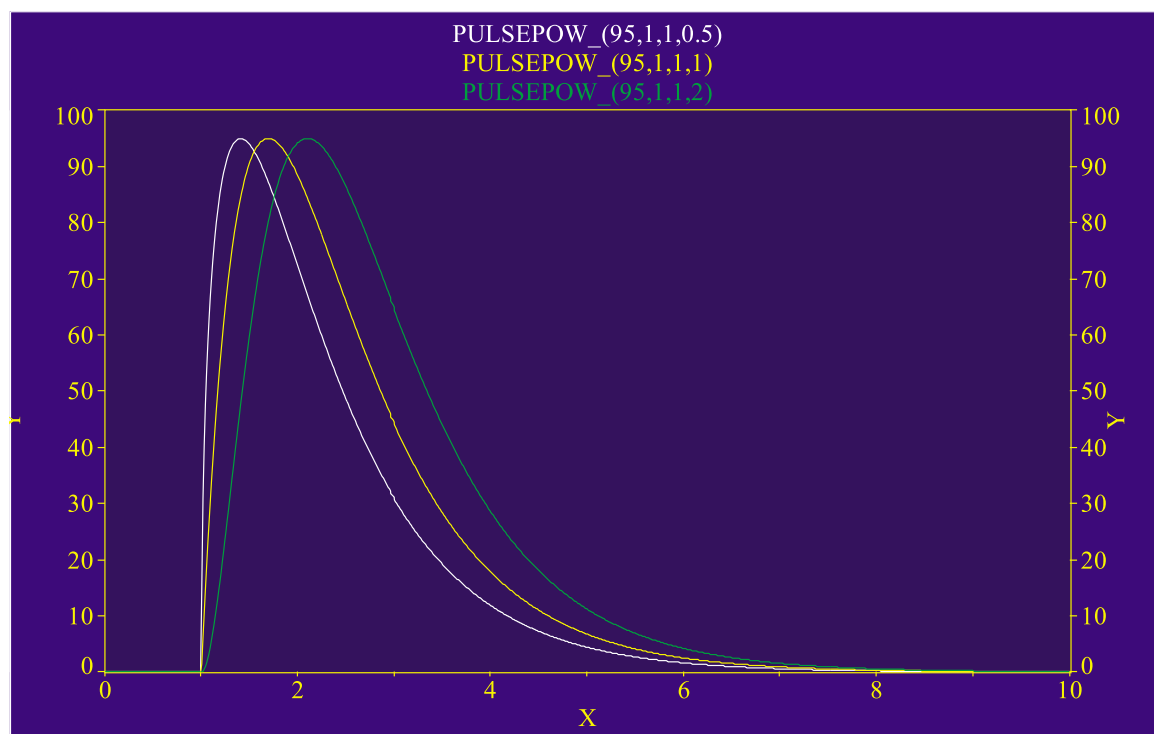
Area: $ac(d+1)^{\frac{d+1}{d}}$

FWHM: Analytic Solution Unknown

Constraints: $c \neq 0, d > 1$

Built-in Function: **LGSTCPOWPK_(a,b,c,d)**

Pulse Peak Modified with Power Term



$$y = \frac{a \left[1 - \exp\left(-\frac{x-b}{c}\right) \right]^d \exp\left(-\frac{x-b}{c}\right)}{d^d (d+1)^{-d-1}}$$

Non-Intercept Form: Equation **8064**

Intercept Form: Equation **8065**

Amplitude: a

Pulse Initiation: b

X at Max: $b + c \ln(1+d)$

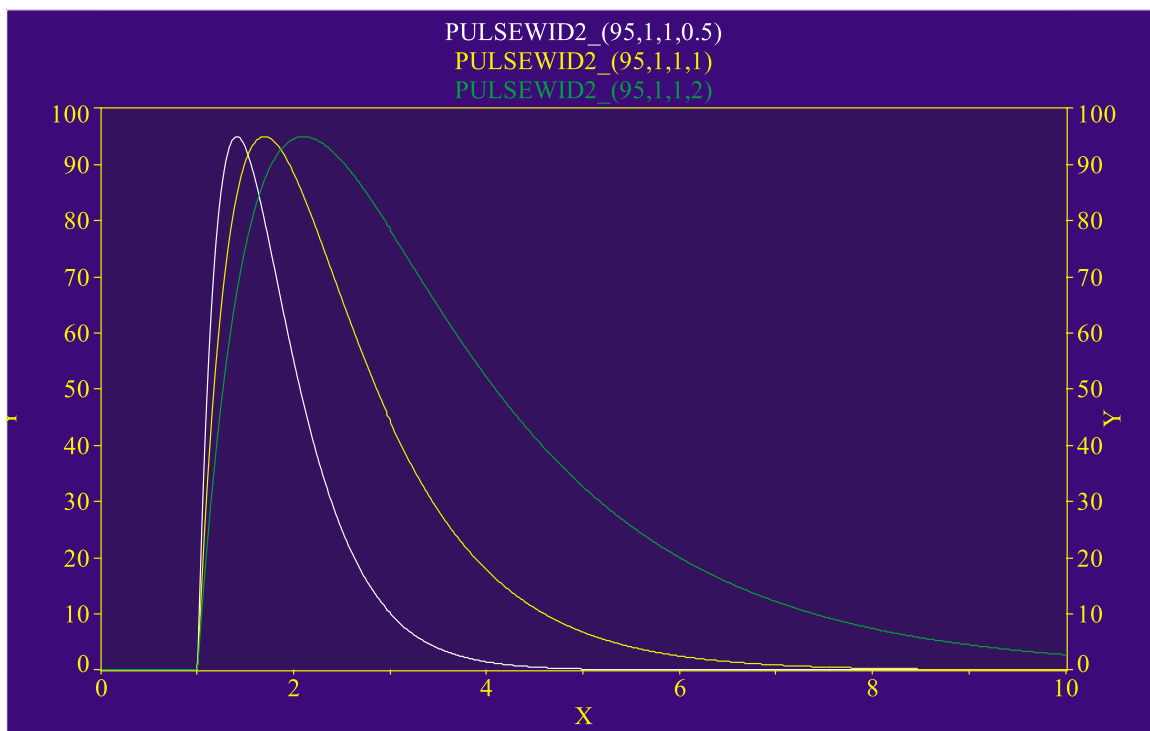
$$\text{Area: } \frac{ac}{(d+1)d^d (d+1)^{-d-1}}$$

FWHM: Analytic Solution Unknown

Constraints: $x \geq b$, $c > 0$, $d > 0$

Built-in Function: **PULSEPOW_(a,b,c,d)**

Pulse Peak with Second Width Term



$$y = \frac{a(c+d) \left[1 - \exp\left(-\frac{x-b}{c}\right) \right] \exp\left(-\frac{x-b}{d}\right)}{d \exp\left(-\frac{c(\ln(c+d) - \ln c)}{d}\right)}$$

Non-Intercept Form: Equation **8066**

Intercept Form: Equation **8067**

Amplitude: a

Pulse Initiation: b

X at Max: $b + c \ln(c+d) - c \ln c$

Area: $ad(c+d)^{\frac{c}{d}} c^{-\frac{c}{d}}$

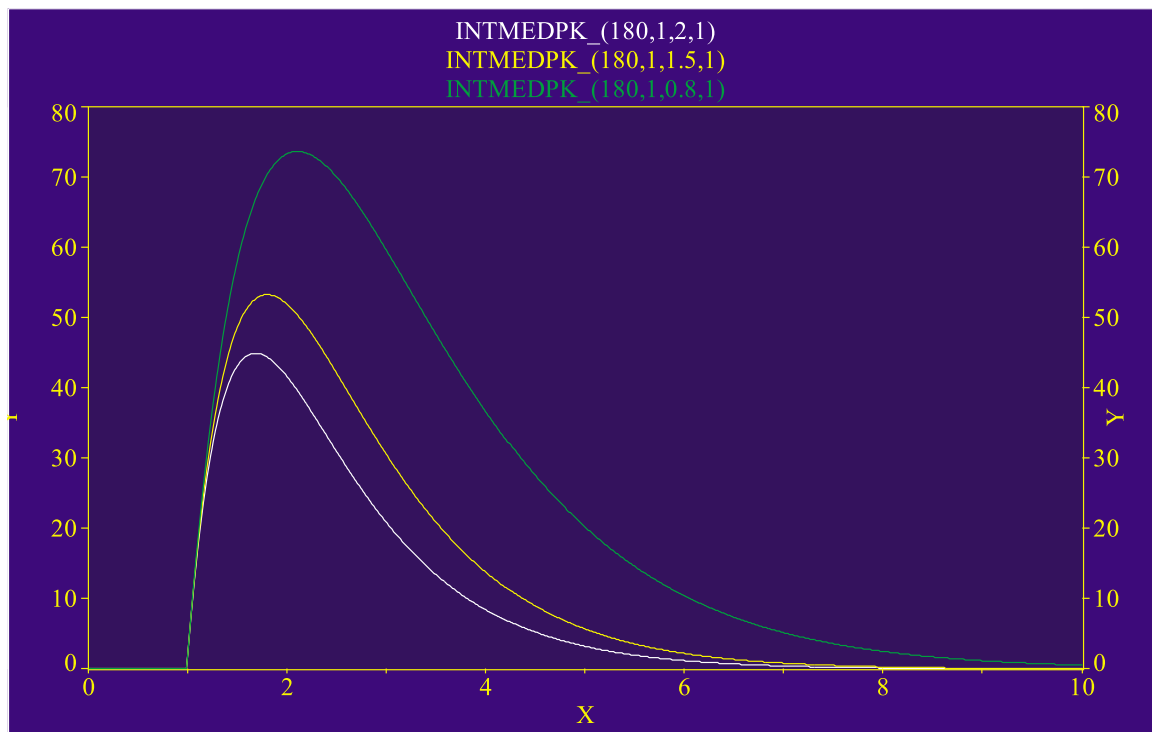
FWHM: Analytic Solution Unknown

Constraints: $x \geq b$, $c > 0$, $d > 0$

Built-in Function: **PULSEWID2_(a,b,c,d)**

Intermediate Peak

A→B→C
Concentration of B



$$y = \frac{ab[\exp(-b(x-d)) - \exp(-c(x-d))]}{c-b}$$

Non-Intercept Form: Equation **8068**

Intercept Form: Equation **8069**

Initial Concentration of A: a

Reaction Begins at: $x=d$

Amplitude (Maximum Concentration of B): $\frac{ab}{c-b} \left[\exp\left(\frac{b \ln b - b \ln c}{c-b}\right) - \exp\left(\frac{c \ln b - c \ln c}{c-b}\right) \right]$

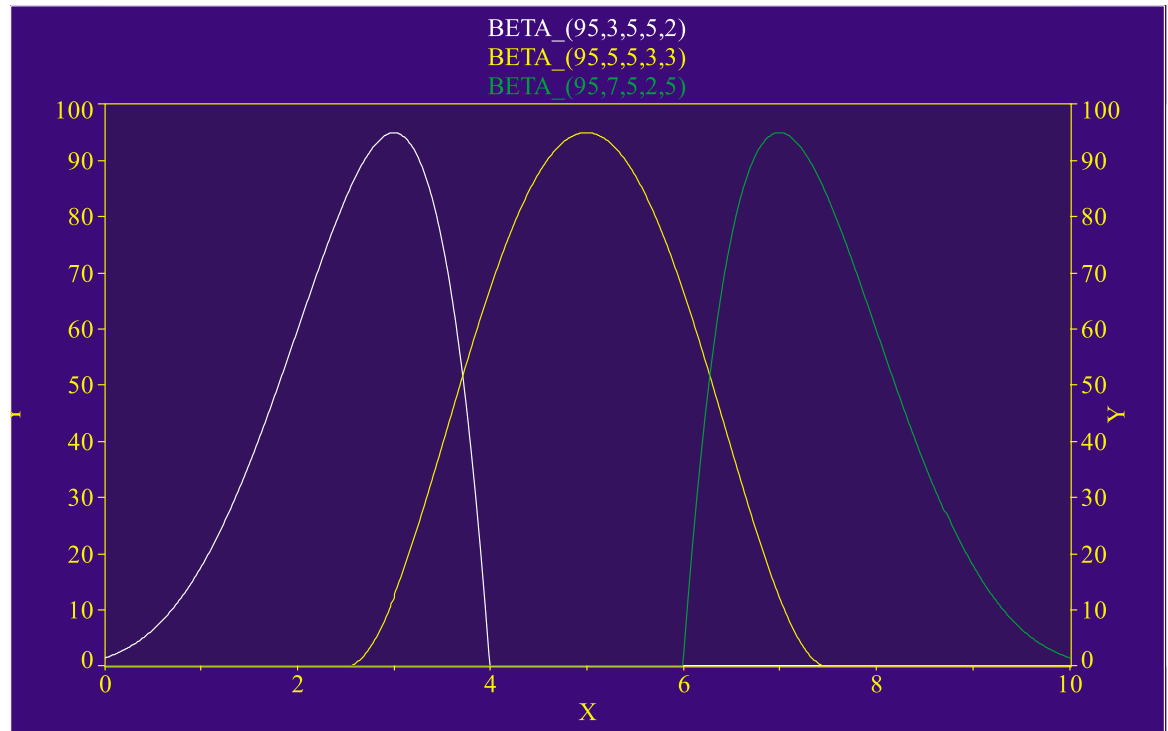
X at Max: $d + \frac{\ln c - \ln b}{c-b}$

Area: $\frac{a}{c}$

Constraints: $x > d$, $b > 0$, $c > 0$, $b \neq c$

Built-in Function: **INTMEDPK_(a,b,c,d)**

Beta



$$y = \frac{a \left(\frac{x - b + \frac{c(d-1)}{d+e-2}}{c} \right)^{d-1} \left(1 - \frac{x - b + \frac{c(d-1)}{d+e-2}}{c} \right)^{e-1}}{\left(\frac{d-1}{d+e-2} \right)^{d-1} \left(\frac{e-1}{d+e-2} \right)^{e-1}}$$

Non-Intercept Form: Equation **8054**

Intercept Form: Equation **8055**

Amplitude: a

X at Max: b

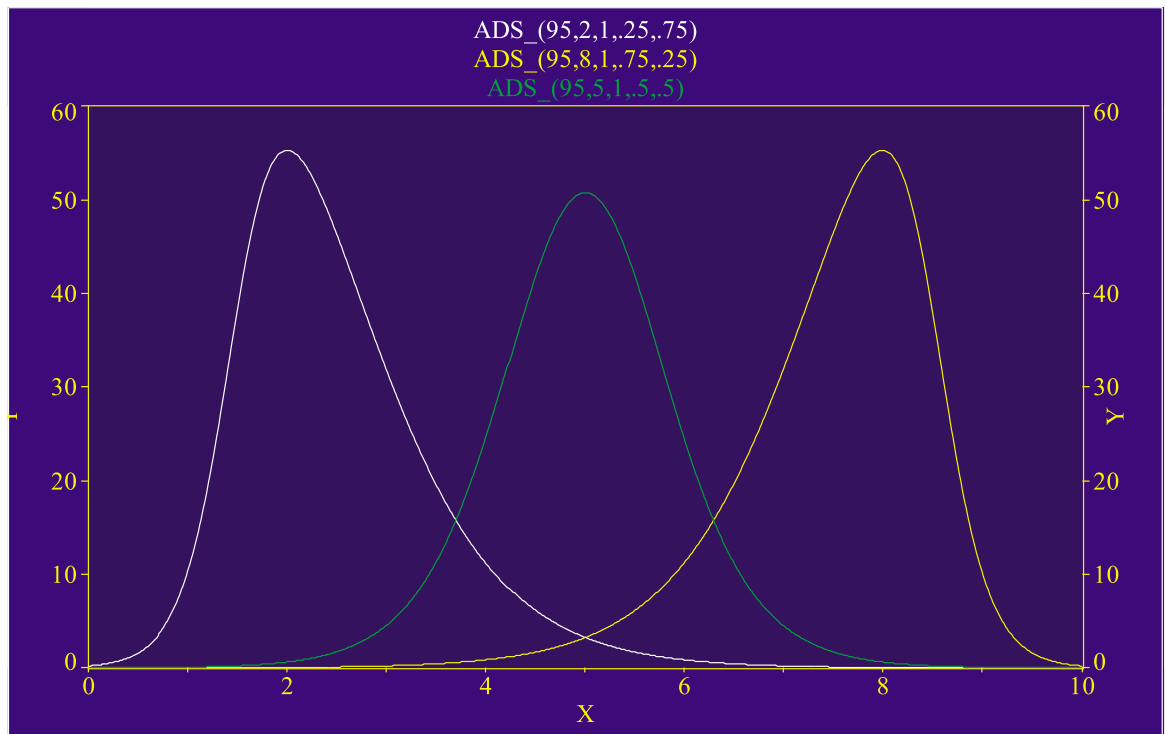
$$\text{Area: } \frac{ac\Gamma(d)\Gamma(e)}{\Gamma(d+e)(d-1)^{d-1}(d+e-2)^{-d-e+2}(e-1)^{e-1}}$$

FWHM: Analytic Solution Unknown

$$\text{Constraints: } \frac{bd + be - 2b - cd + c}{d + e - 2} \leq x \leq \frac{bd + be - 2b + ce - c}{d + e - 2}, c > 0, d > 1, e > 1$$

Built-in Function: **BETA_(a,b,c,d,e)**

Asymmetric Double Sigmoidal (ADS)



$$y = \left[\frac{a}{1 + \exp\left(-\frac{x - b + c/2}{d}\right)} \right] \left[1 - \frac{1}{1 + \exp\left(-\frac{x - b - c/2}{e}\right)} \right]$$

Non-Intercept Form: Equation **8056**

Intercept Form: Equation **8057**

Maximum Amplitude: a

X at Max: Analytic Solution Unknown

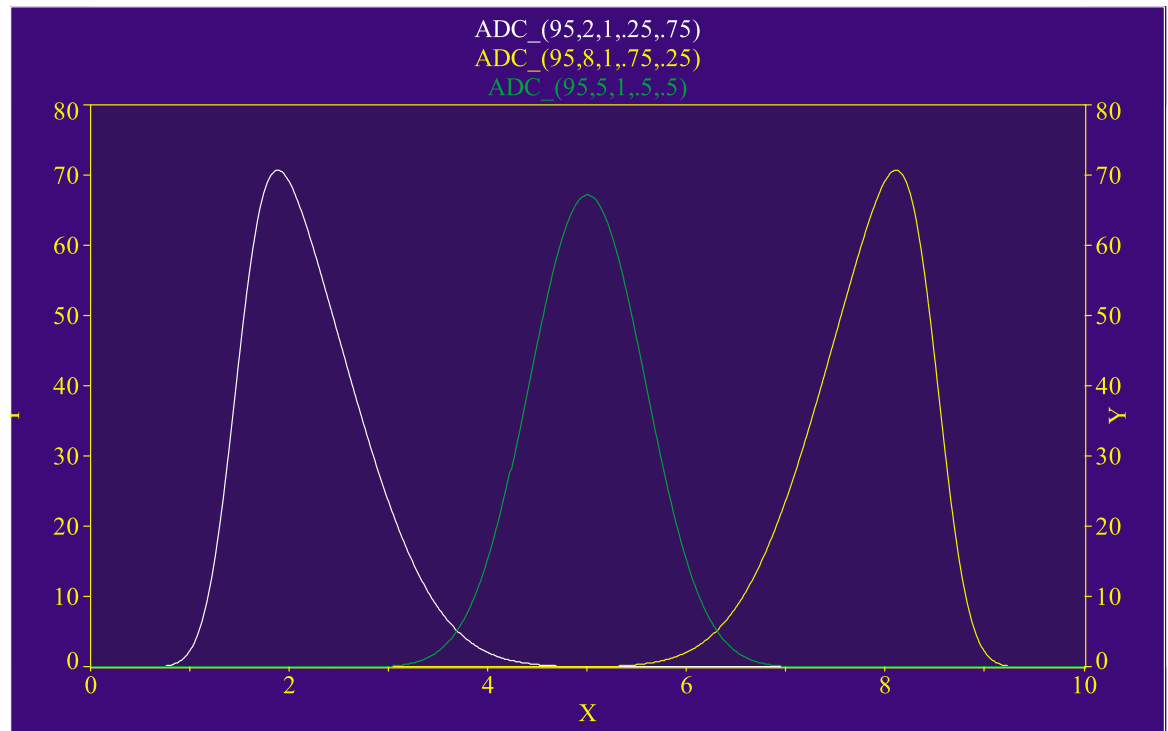
Area: Analytic Solution Unknown

FWHM: Analytic Solution Unknown

Constraints: $c > 0, d > 0, e > 0$

Built-in Function: **ADS_(a,b,c,d,e)**

Asymmetric Double Gaussian Cumulative (ADC)



$$y = \frac{a}{2} \left[1 + \operatorname{erf} \left(\frac{x - b + c/2}{\sqrt{2}d} \right) \right] \left[\frac{1}{2} - \frac{1}{2} \operatorname{erf} \left(\frac{x - b - c/2}{\sqrt{2}e} \right) \right]$$

Non-Intercept Form: Equation **8058**

Intercept Form: Equation **8059**

Maximum Amplitude: a

X at Max: Analytic Solution Unknown

Area: Analytic Solution Unknown

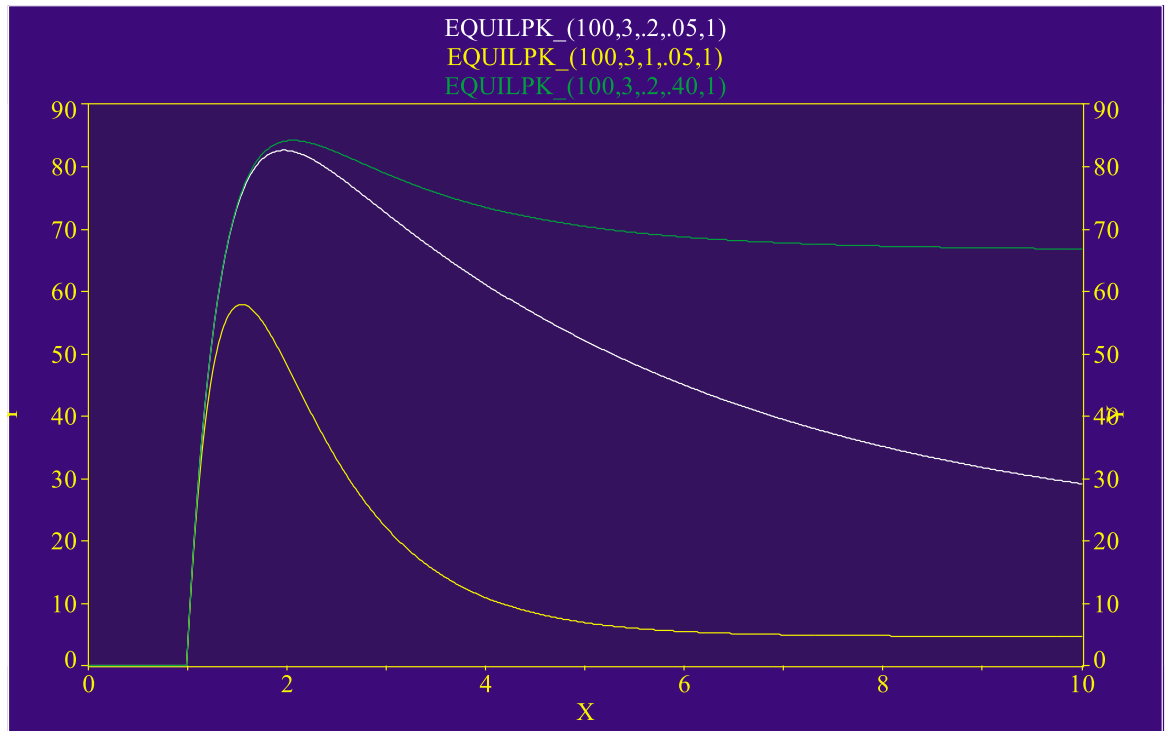
FWHM: Analytic Solution Unknown

Constraints: $c > 0$, $d > 0$, $e > 0$

Built-in Function: **ADC_(a,b,c,d,e)**

Equilibrium Peak

$A \leftrightarrow B \leftrightarrow C$
Concentration of B



$$y = a \left[1 - \exp(-b(x-e)) - \frac{c}{c+d} \left(1 + \frac{b \exp(-(c+d)(x-e)) - (c+d) \exp(-b(x-e))}{c+d-b} \right) \right]$$

Non-Intercept Form: Equation **8070**

Intercept Form: Equation **8071**

Initial Conc of A: a

Reaction Begins at: $x=d$

$$X \text{ at Max: } x_c = e + \frac{\ln(bc(-c-d)) - \ln(b(c+d)(d-b))}{c+d-b}$$

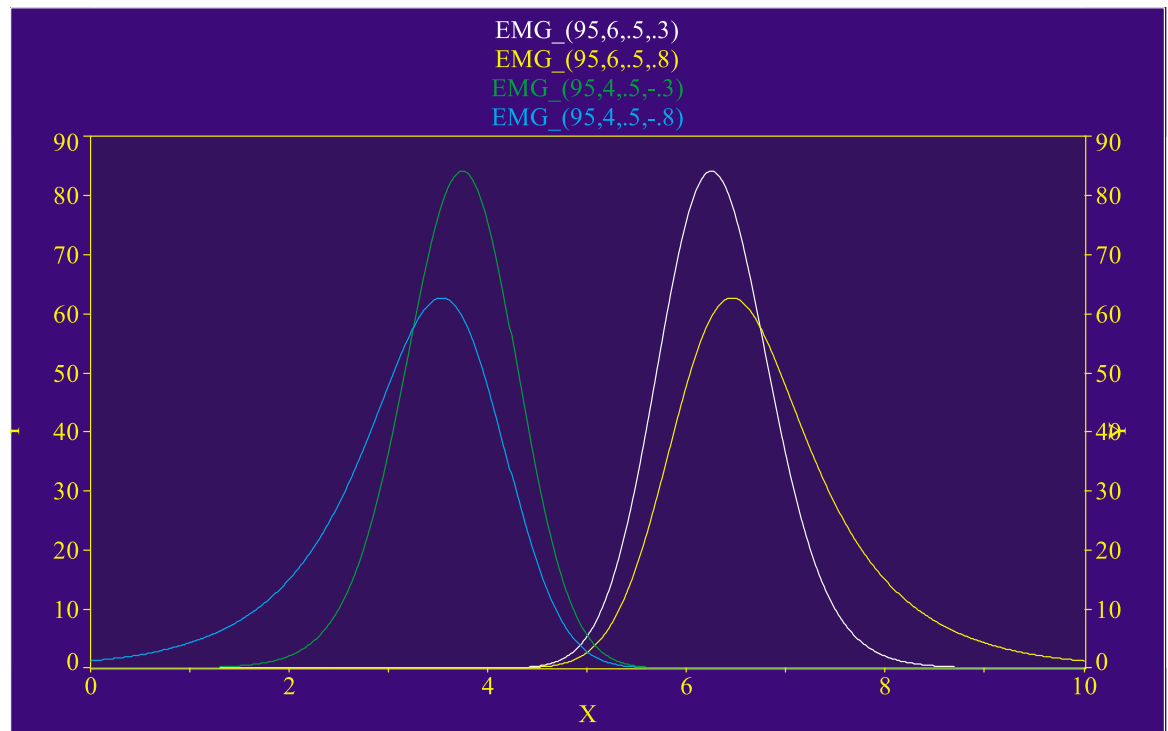
$$\text{Amplitude (Max Conc): } a \left[1 - \exp(-bx_c) - \frac{c}{c+d} \left(1 + \frac{b \exp(-(c+d)x_c) - (c+d) \exp(-bx_c)}{c+d-b} \right) \right]$$

$$\text{Equilibrium Conc: } a \left(1 - \frac{c}{c+d} \right)$$

Constraints: $x > e$, $b > 0$, $c > 0$, $d > 0$, $(c+d) \neq b$

Built-in Function: **EQUILPK_(a,b,c,d,e)**

Exponentially Modified Gaussian (EMG)



$$y = \frac{ac\sqrt{2\pi}}{2d} \exp\left(\frac{b-x}{d} + \frac{c^2}{2d^2}\right) \left[\frac{d}{|d|} - \operatorname{erf}\left(\frac{b-x}{\sqrt{2}c} + \frac{c}{\sqrt{2}d}\right) \right]$$

Non-Intercept Form: Equation **8164**

Intercept Form: Equation **8165**

Amplitude, deconvolved Gaussian: a

Center, deconvolved Gaussian: b

Area, deconvolved Gaussian or function: $\sqrt{2\pi}ac$

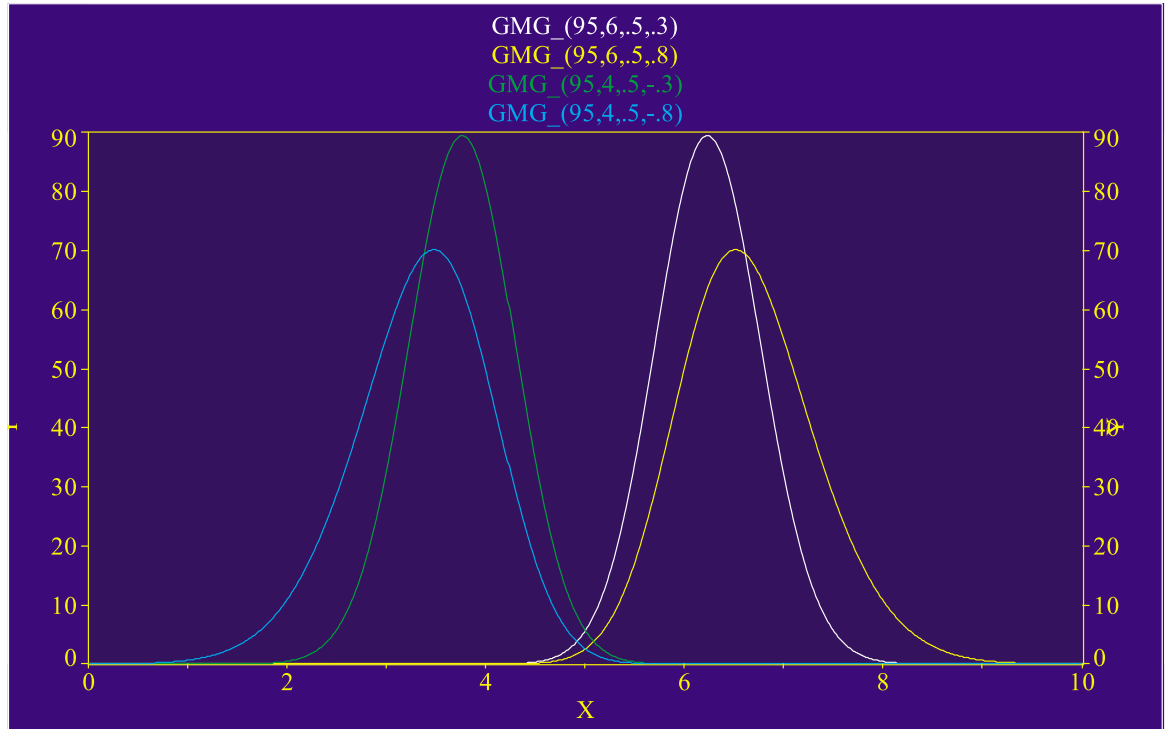
FWHM, deconvolved Gaussian: $2\sqrt{2\ln 2}c$

Time Constant, Exponential: d

Constraints: $c>0, d\neq 0$

Built-in Function: **EMG_(a,b,c,d)**

Half-Gaussian Modified Gaussian (GMG)



$$y = \frac{ac}{\sqrt{d^2 + c^2}} \exp\left(-\frac{1}{2} \frac{(x-b)^2}{d^2 + c^2}\right) \left[1 + \operatorname{erf}\left(\frac{1}{2} \frac{\sqrt{2}d(x-b)}{c\sqrt{d^2 + c^2}}\right)\right]$$

Non-Intercept Form: Equation **8166**

Intercept Form: Equation **8167**

Amplitude, deconvolved Gaussian: a

Center, deconvolved Gaussian: b

Area, deconvolved Gaussian or function:

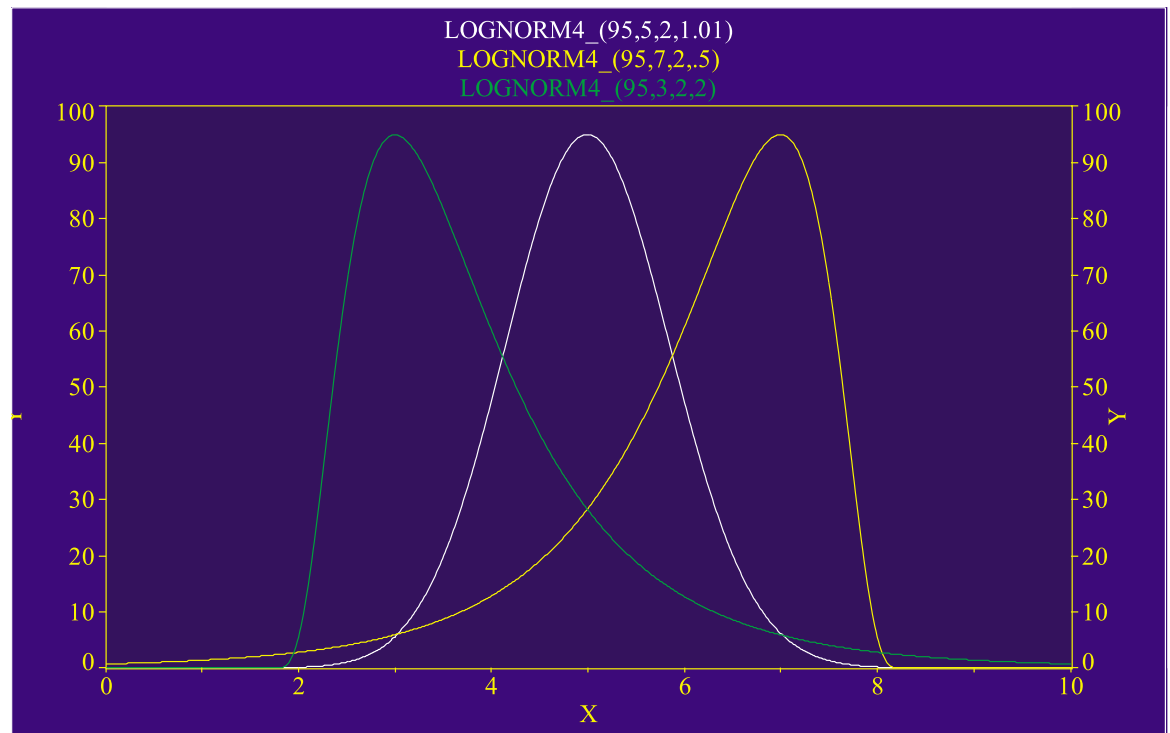
FWHM, deconvolved Gaussian:

Half-Gaussian Response Width: d

Constraints: $c > 0$, $d \neq 0$

Built-in Function: **GMG_(a,b,c,d)**

Log Normal 4-Parameter



$$y = a \exp \left[-\frac{\ln(2)}{\ln(d)^2} \ln \left(\frac{(x-b)(d^2-1)}{cd} + 1 \right)^2 \right]$$

Non-Intercept Form: Equation **8174**

Intercept Form: Equation **8175**

Amplitude: a

X at Max: b

$$\text{Area: } \frac{\sqrt{\pi}acd \ln d \exp \left(\frac{\ln(d)^2}{4 \ln 2} \right)}{\sqrt{\ln 2}(d^2 - 1)}$$

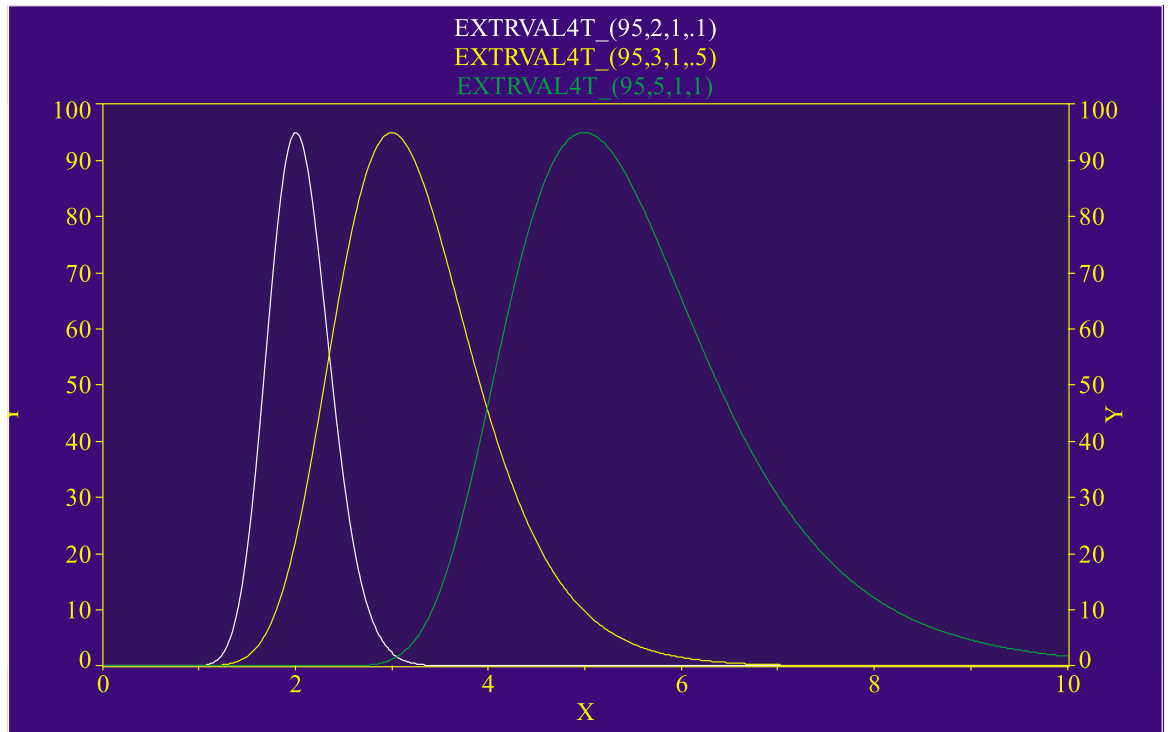
FWHM: c

Asymmetry at half-maximum: d

Constraints: $x > b - \frac{cd}{d^2 - 1}$ if $d > 1$, $x < b - \frac{cd}{d^2 - 1}$ if $d < 1$, $c > 0$, $d > 0$, $d \neq 1$

Built-in Function: **LOGNORM4_(a,b,c,d)**

Extreme Value 4-Parameter Tailed



$$y = a \exp \left[\frac{-x + b + c - cd \exp \left(-\frac{x + c \ln d - b}{c} \right)}{cd} \right]$$

Non-Intercept Form: Equation **8176**

Intercept Form: Equation **8177**

Amplitude: a

X at Max: b

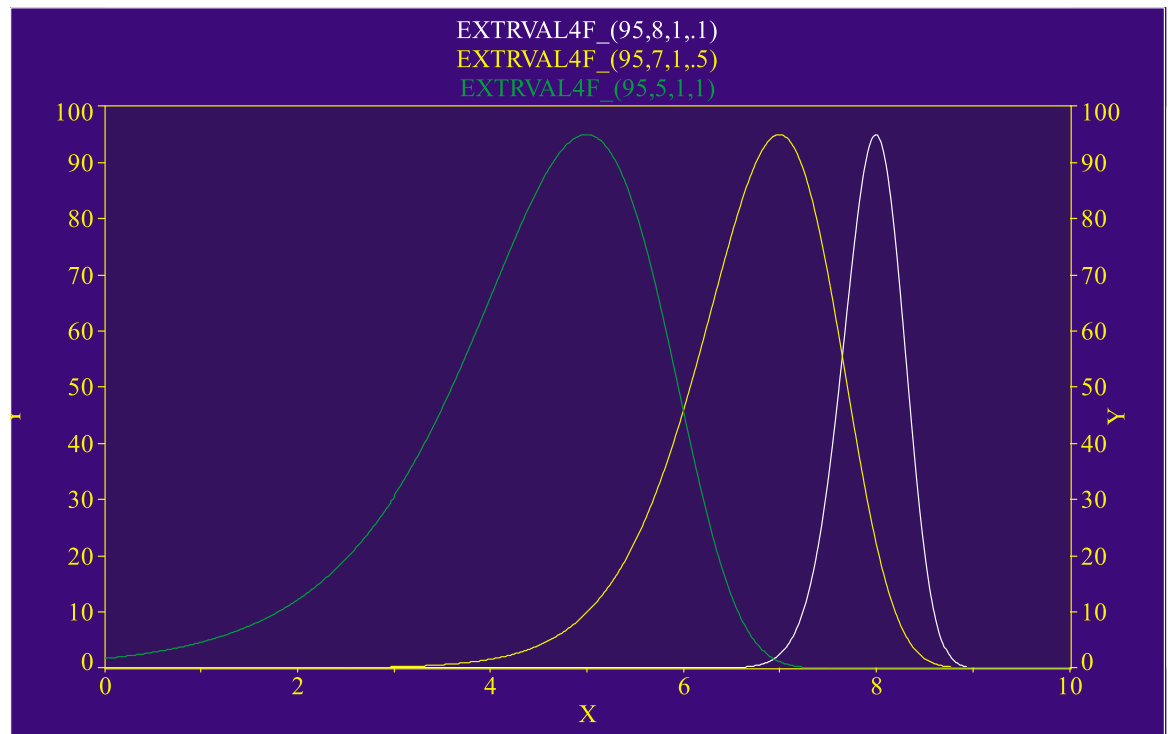
Area: $acd^{\frac{1}{d}} \exp \left(\frac{1}{d} \right) \Gamma \left(\frac{1}{d} \right)$

FWHM: Analytic Solution Unknown

Constraints: $c > 0, d > 0.01$

Built-in Function: **EXTRVAL4T_(a,b,c,d)**

Extreme Value 4-Parameter Fronted



$$y = a \exp \left[\frac{x + c - b - cd \exp \left(\frac{x - c \ln d - b}{c} \right)}{cd} \right]$$

Non-Intercept Form: Equation **8178**

Intercept Form: Equation **8179**

Amplitude: a

X at Max: b

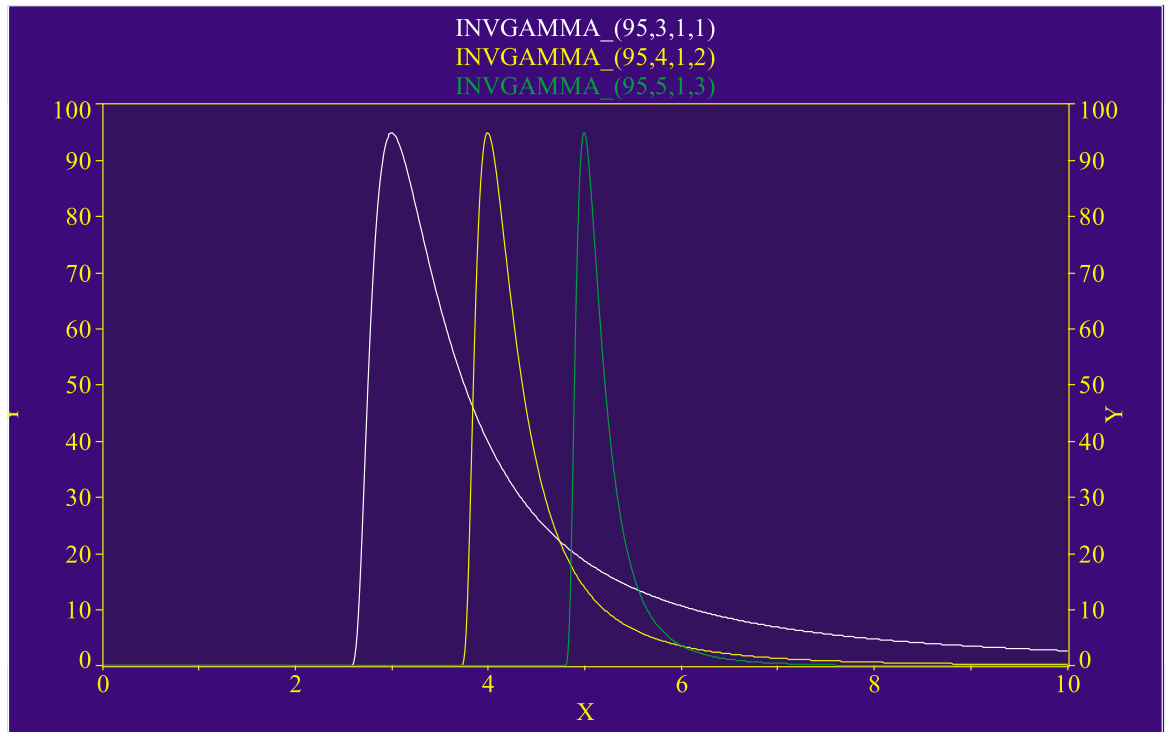
Area: $ac \exp \left(\frac{1 + \ln d}{d} \right) \Gamma \left(\frac{1}{d} \right)$

FWHM: Analytic Solution Unknown

Constraints: $c > 0, d > 0.01$

Built-in Function: **EXTRVAL4F_(a,b,c,d)**

Inverted Gamma



$$y = \frac{ac \exp\left(\frac{(x-b)(d+1)^2}{(d+1)(x-b)+c}\right) \left(\frac{(d+1)(x-b)+c}{c}\right)^{-d}}{(d+1)(x-b)+c}$$

Non-Intercept Form: Equation **8180**

Intercept Form: Equation **8181**

Amplitude: a

X at Max: b

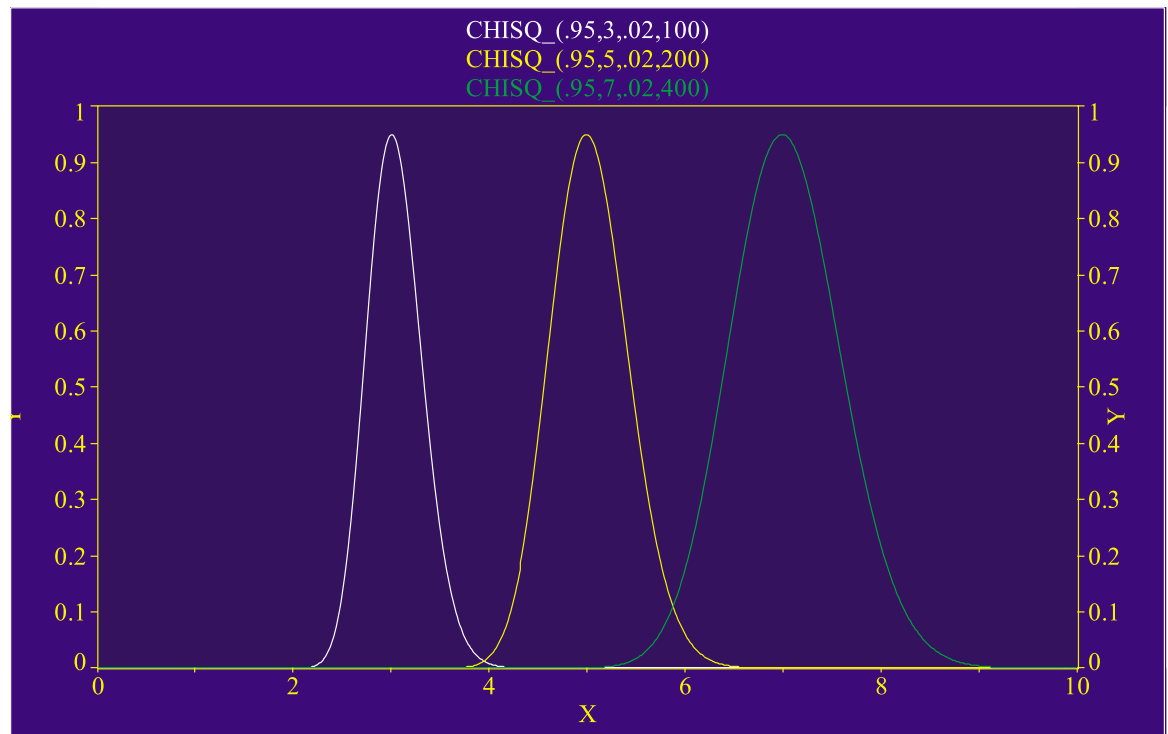
Area: $ac \exp(d+1)(d+1)^{-d-1} \Gamma(d)$

FWHM: Analytic Solution Unknown

Constraints: $x > \frac{b-c+bd}{d+1}, c > 0, d > 0$

Built-in Function: **INV GAMMA_(a,b,c,d)**

Chi-Squared



$$y = \frac{a \left(\frac{x - b + c(d-2)}{c} \right)^{d/2-1} \exp \left(-\frac{1}{2} \frac{x - b + c(d-2)}{c} \right)}{(d-2)^{d/2-1} \exp \left(-\frac{d}{2} + 1 \right)}$$

Non-Intercept Form: Equation **8182**

Intercept Form: Equation **8183**

Amplitude: a

X at Max: b

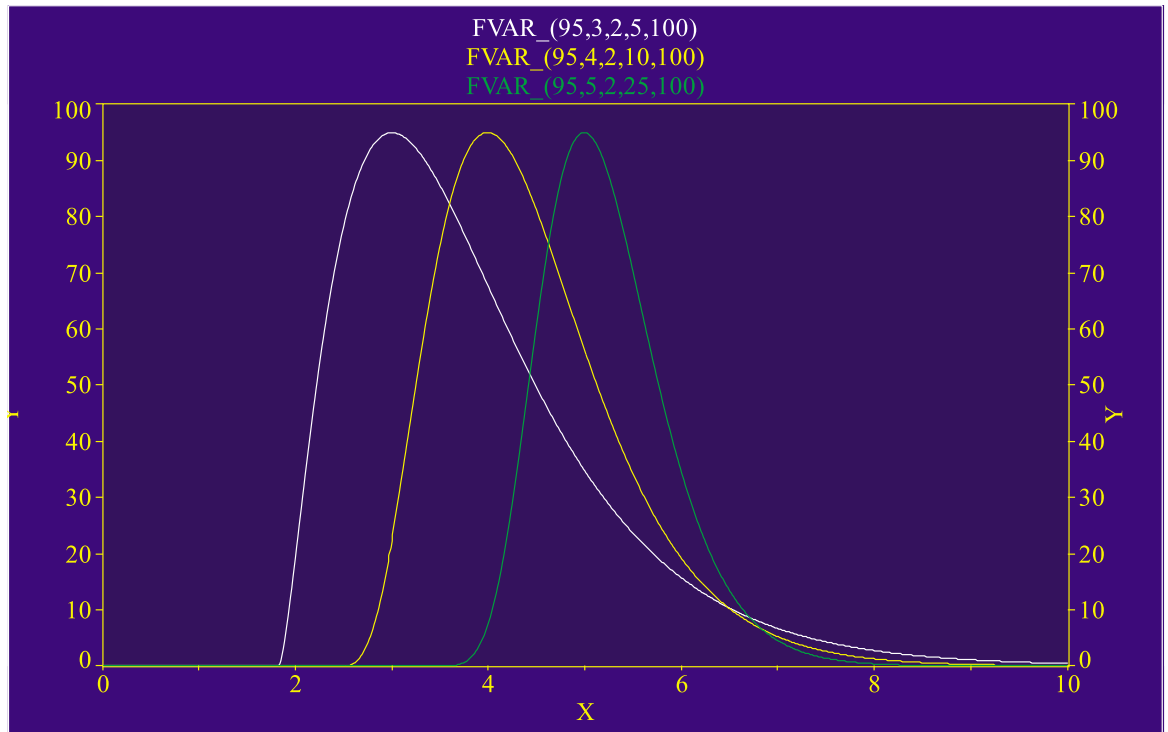
$$\text{Area: } \frac{ac2^{d/2} \Gamma \left(\frac{d}{2} \right)}{(d-2)^{d/2-1} \exp \left(-\frac{d}{2} + 1 \right)}$$

FWHM: Analytic Solution Unknown

Constraints: $x > b - cd + 2c, c > 0, d > 2$

Built-in Function: **CHISQ_(a,b,c,d)**

F-Variance



$$y = \frac{a \left(\frac{x-b}{c} + \frac{e(d-2)}{d(e+2)} \right)^{d/2-1} \left(1 + \frac{d-2}{e+2} \right)^{d/2+e/2}}{\left[1 + \frac{d}{e} \left(\frac{x-b}{c} + \frac{e(d-2)}{d(e+2)} \right) \right]^{d/2+e/2} \left(\frac{e(d-2)}{d(e+2)} \right)^{d/2-1}}$$

Non-Intercept Form: Equation **8184**Intercept Form: Equation **8185**Amplitude: a X at Max: b

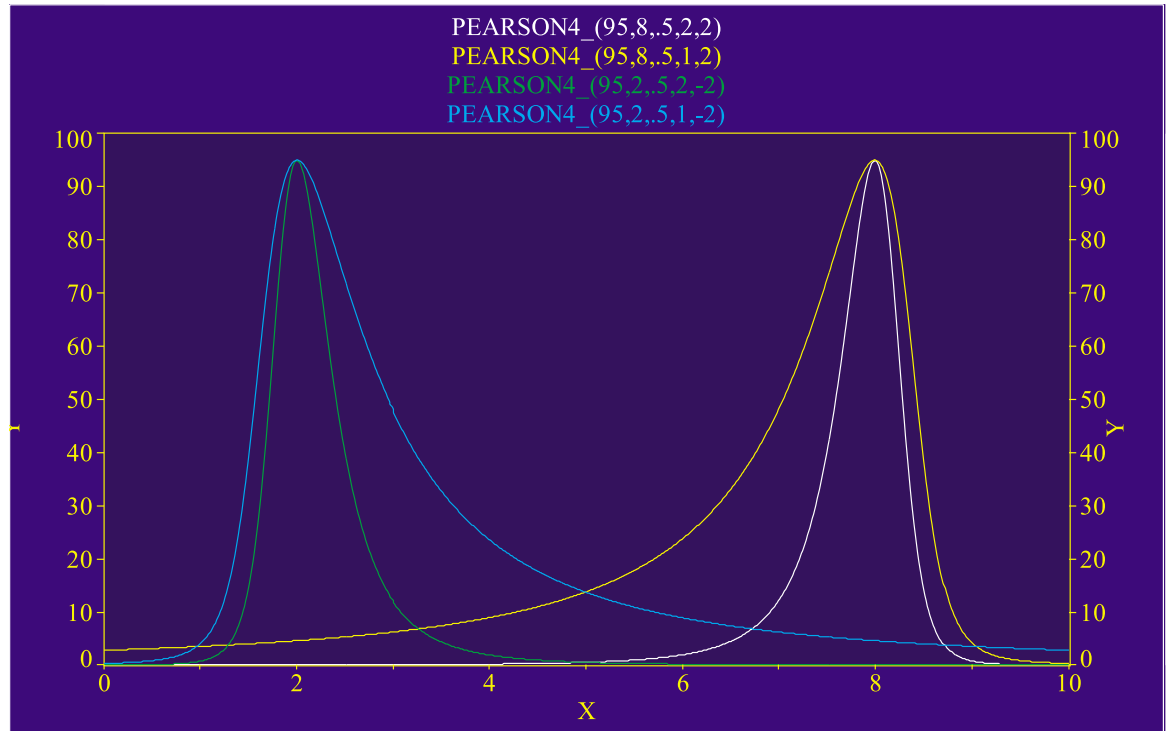
$$\text{Area: } \frac{ace(d-2) \left(\frac{d+e}{2+e} \right)^{d/2+e/2} \Gamma\left(\frac{d}{2}\right) \Gamma\left(\frac{e}{2}\right) \left(\sqrt{\frac{d-2}{e+2}} \right)^{-d}}{d(e+2) \Gamma\left(\frac{d}{2} + \frac{e}{2}\right)}$$

FWHM: Analytic Solution Unknown

$$\text{Constraints: } x > b - \frac{ce(d-2)}{d(e+2)}, c > 0, d > 2, e > 2$$

Built-in Function: **FVAR_(a,b,c,d,e)**

Pearson IV



$$y = a \frac{\left[1 + \frac{\left(x - \frac{ce}{2d} - b \right)^2}{c^2} \right]^{-d} \exp \left[-e \left(\tan^{-1} \left(\frac{x - \frac{ce}{2d} - b}{c} \right) + \tan^{-1} \left(\frac{e}{2d} \right) \right) \right]}{\left(1 + \frac{e^2}{4d^2} \right)^{-d}}$$

Non-Intercept Form: Equation **8186**

Intercept Form: Equation **8187**

Amplitude: a

X at Max: b

Area: Analytic Solution Unknown

FWHM: Analytic Solution Unknown

Constraints: $c > 0$, $d > 0$

Built-in Function: **PEARSON4_(a,b,c,d,e)**

Symmetric Transition Functions

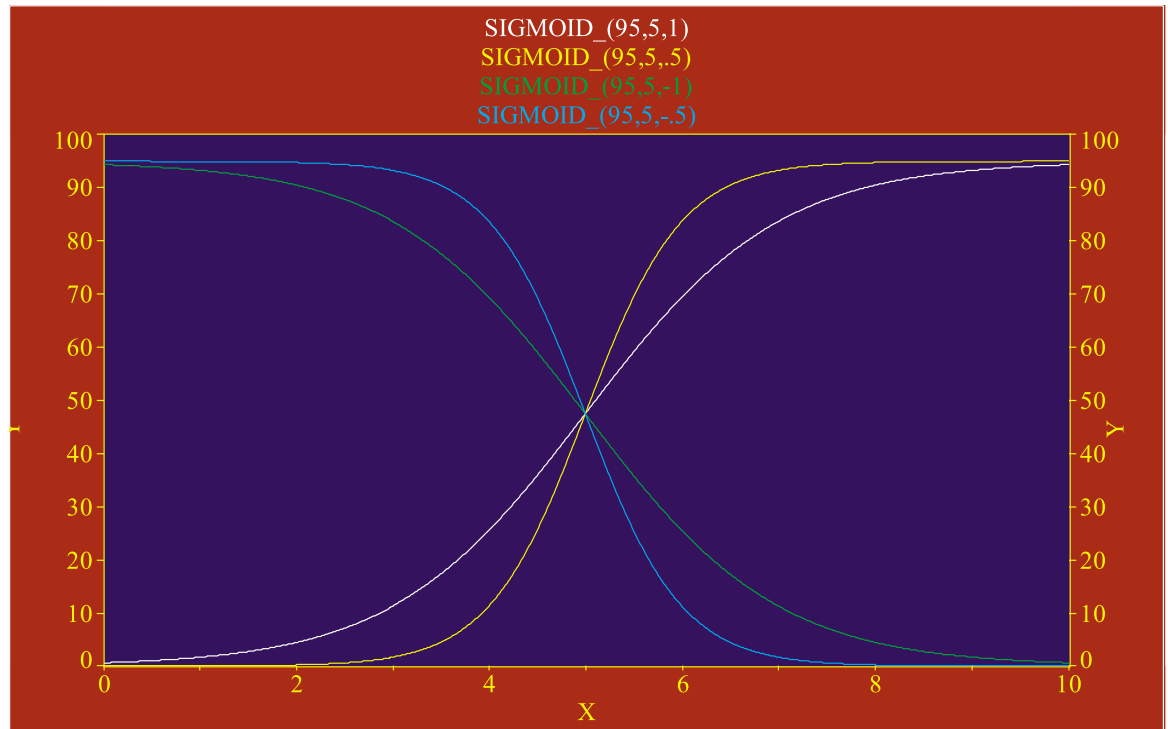
These transition functions exhibit a symmetry about the transition center in both the X and Y dimensions. In TableCurve 2D, the transition width is the X distance between $Y_{\min} + .75 * Y_{\text{range}}$ and $Y_{\min} + .25 * Y_{\text{range}}$. This width will be positive for an ascending transition and negative for a descending transition.

Some transition functions will produce the opposite direction transition simply by changing the sign on the width term. When this property exists, TableCurve 2D will use this to enable the fitting of descending transitions with the zero-intercept version of the function. When this property is absent, descending transitions will require the intercept form. When a function can model a descending transition with a negative width term, such will be illustrated in the accompanying graph.

Unless otherwise indicated, a represents the transition magnitude or height, b represents the midpoint of the transition (the X where $Y = Y_{\min} + .5 * Y_{\text{range}}$), c controls the width of the transition, and d , if present, controls the shape of the transition.

- Sigmoid
- Gaussian Cumulative
- Lorentzian Cumulative
- Cumulative SDS (Symmetric Double Sigmoidal)

Sigmoid



$$y = \frac{a}{1 + \exp\left(-\frac{x - b}{c}\right)}$$

Non-Intercept Form: Equation **8074**

Intercept Form: Equation **8011**

Transition Height: a

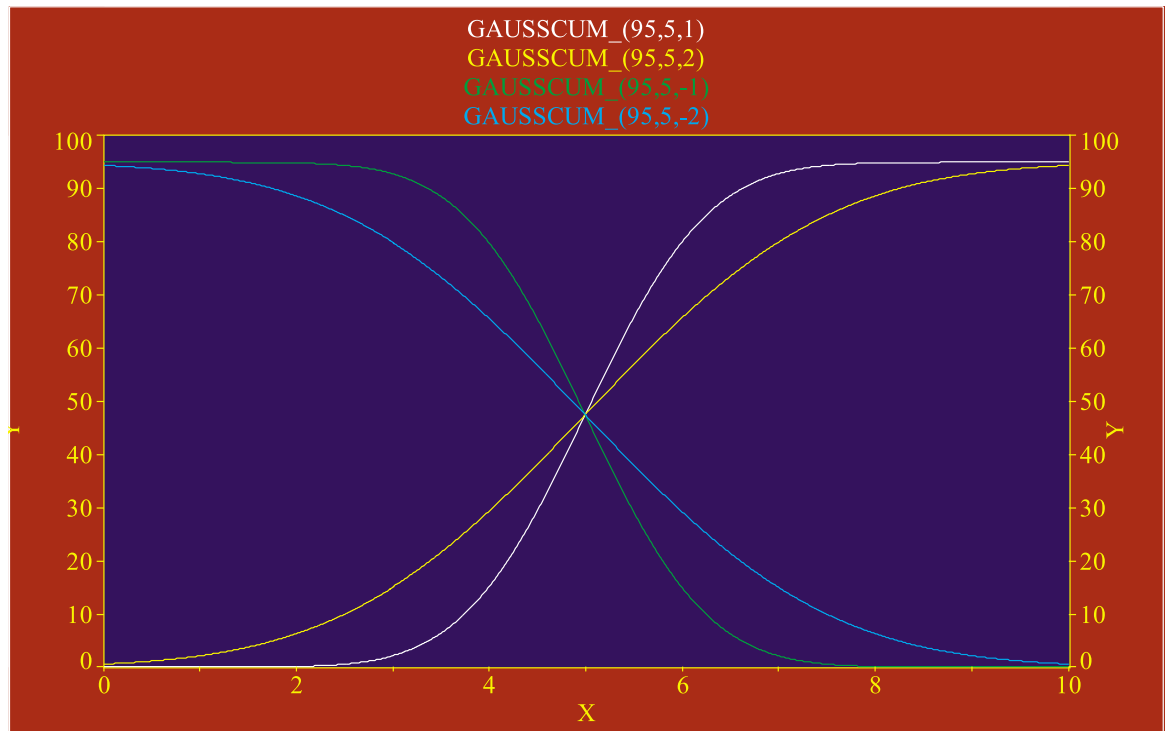
Transition Center: b

Transition Width: $2.197224578c$

Constraints: $c \neq 0$

Built-in Function: **SIGMOID_(a,b,c)**

Gaussian Cumulative



$$y = \frac{a}{2} \left[1 + \operatorname{erf} \left(\frac{x - b}{\sqrt{2}c} \right) \right]$$

Non-Intercept Form: Equation **8075**

Intercept Form: Equation **8012**

Transition Height: a

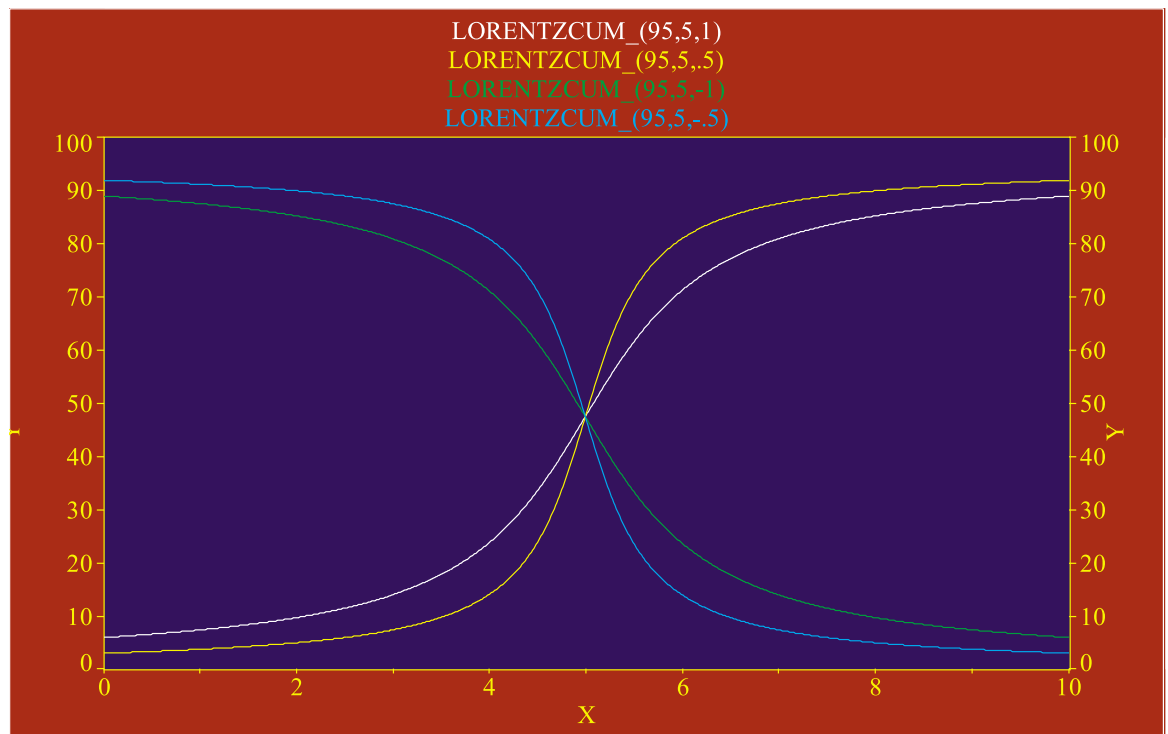
Transition Center: b

Transition Width: $1.348979501c$

Constraints: $c \neq 0$

Built-in Function: **GAUSSCUM_(a,b,c)**

Lorentzian Cumulative



$$y = \frac{a}{\pi} \left[\arctan\left(\frac{x-b}{c}\right) + \frac{\pi}{2} \right]$$

Non-Intercept Form: Equation **8077**

Intercept Form: Equation **8078**

Transition Height: a

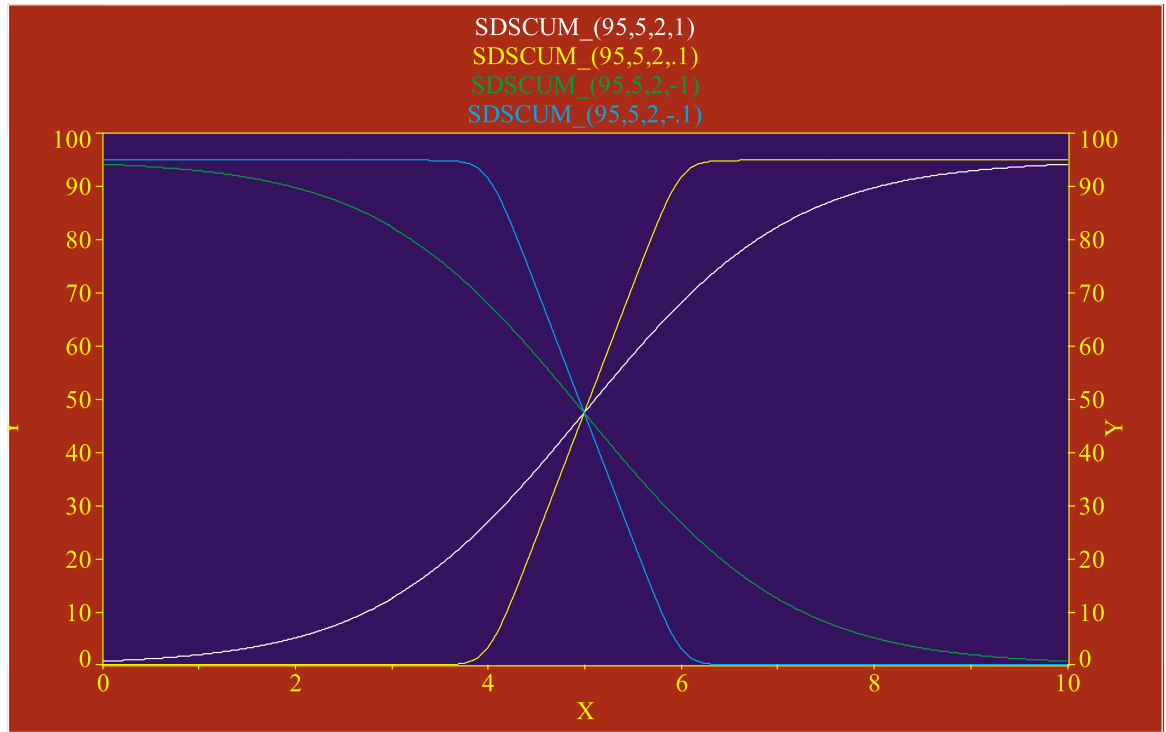
Transition Center: b

Transition Width: $2c$

Constraints: $c \neq 0$

Built-in Function: **LORENTZCUM_(a,b,c)**

Cumulative SDS (Symmetric Double Sigmoidal)



$$y = \frac{a}{2c} \left[2d \ln \left(\exp \left(\frac{2x+c}{2d} \right) + \exp \left(\frac{b}{d} \right) \right) - 2d \ln \left(\exp \left(\frac{x}{d} \right) + \exp \left(\frac{2b+c}{2d} \right) \right) + c \right]$$

Non-Intercept Form: Equation **8093**

Intercept Form: Equation **8094**

Transition Height: a

Transition Center: b

Transition Width:

$$d \ln \left(1 - \exp \left(\frac{3c}{4d} \right) \right) - d \ln \left(\exp \left(\frac{c}{4d} \right) - \exp \left(\frac{c}{2d} \right) \right) - d \ln \left(1 - \exp \left(\frac{c}{4d} \right) \right) + d \ln \left(\exp \left(\frac{-c}{4d} \right) - \exp \left(\frac{c}{2d} \right) \right)$$

Constraints: $c \neq 0$, $d \neq 0$

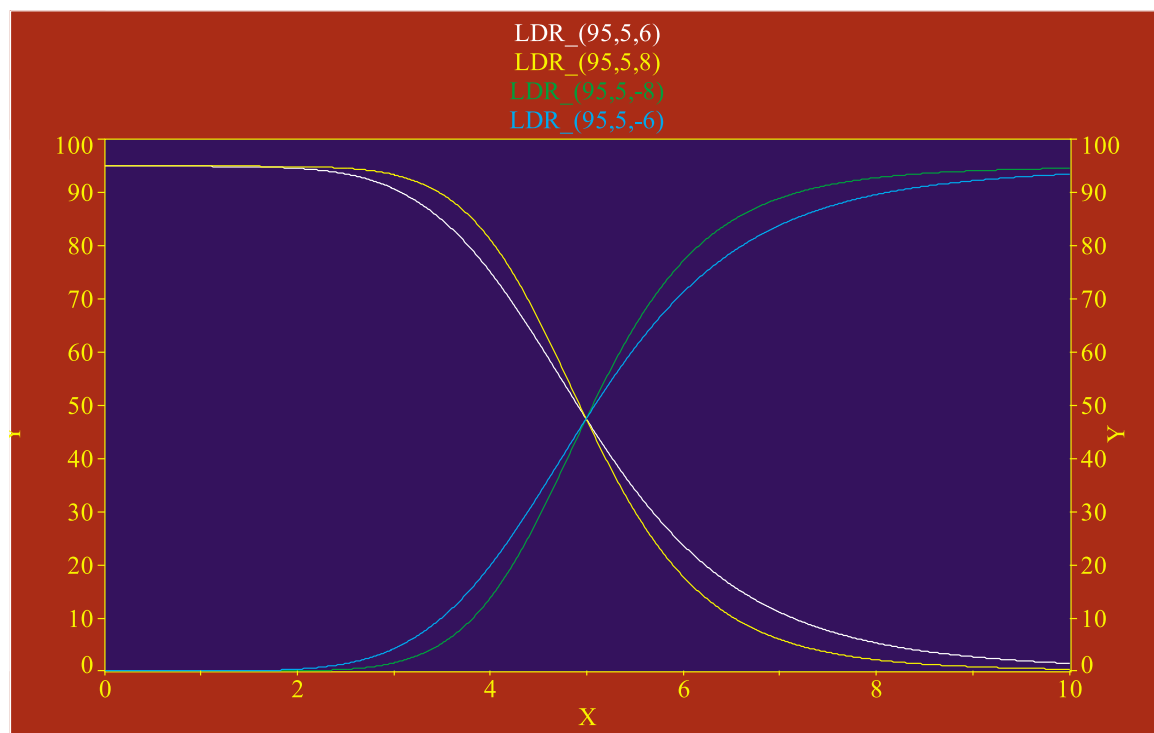
Built-in Function: **SDSCUM_(a,b,c,d)**

Asymmetric Transition Functions

These transition functions normally exhibit an asymmetry about the transition center. For asymmetric functions, this center remains defined as the x at half transition height. The analysis is identical to the symmetric transitions except that any d term present, unless otherwise indicated, controls the degree of asymmetry.

- Logistic Dose Response (LDR)
- Log Normal Cumulative
- Extreme Value Cumulative
- Pulse Cumulative
- Pulse Cumulative with Power Term
- Weibull Cumulative
- Asymmetric Sigmoid
- Asymmetric Sigmoid Reverse Asymmetry
- Cascade Formation
- Impulse Transition
- Cumulative EMG (Exponentially Modified Gaussian)

Logistic Dose Response (LDR)



$$y = \frac{a}{1 + \left(\frac{x}{b}\right)^c}$$

Non-Intercept Form: Equation **8076**

Intercept Form: Equation **8013**

Transition Height: a

Transition Center: b

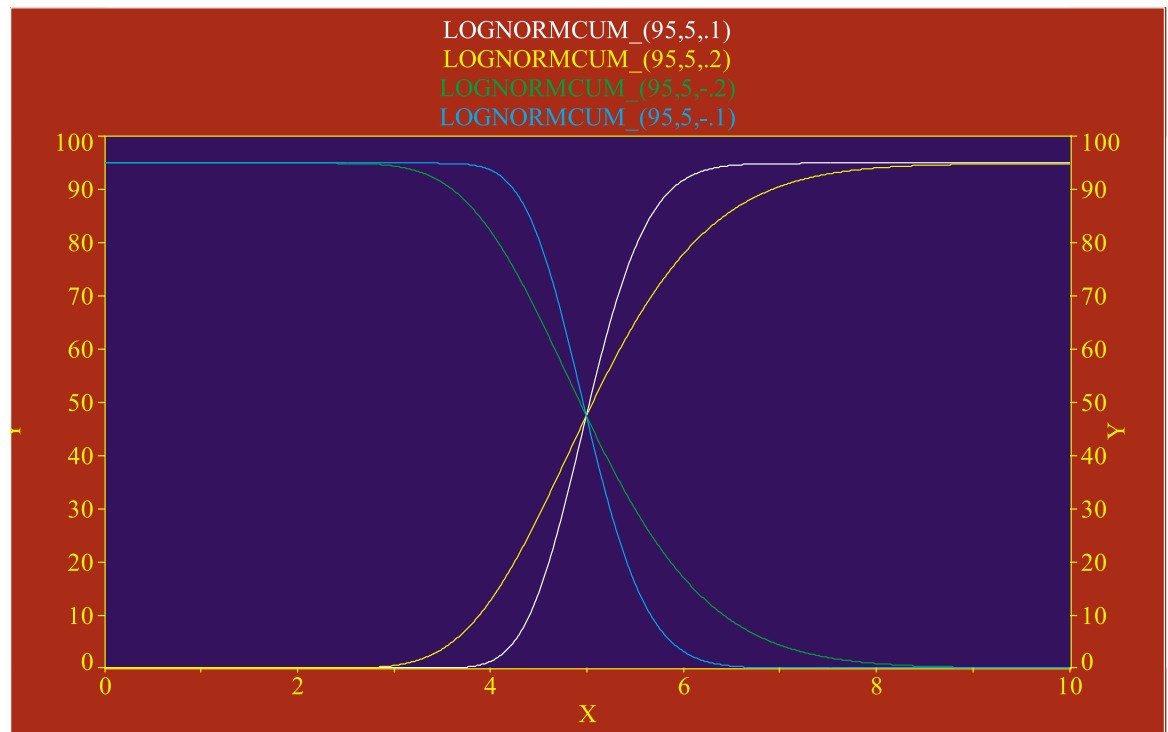
Transition Width: $b \left(3^{\frac{-1}{c}} - 3^{\frac{1}{c}} \right)$

Constraints: $x > 0, b > 0$

Built-in Function: **LDR_(a,b,c)**

Note that TableCurve 2D's LDR function consists of the widely used form where a positive c term produces a descending transition and a negative c term produces the ascending form.

Log Normal Cumulative



$$y = \frac{a}{2} \operatorname{erfc} \left(\frac{-\ln \left(\frac{x}{b} \right)}{\sqrt{2}c} \right)$$

Non-Intercept Form: Equation **8079**

Intercept Form: Equation **8080**

Transition Height: a

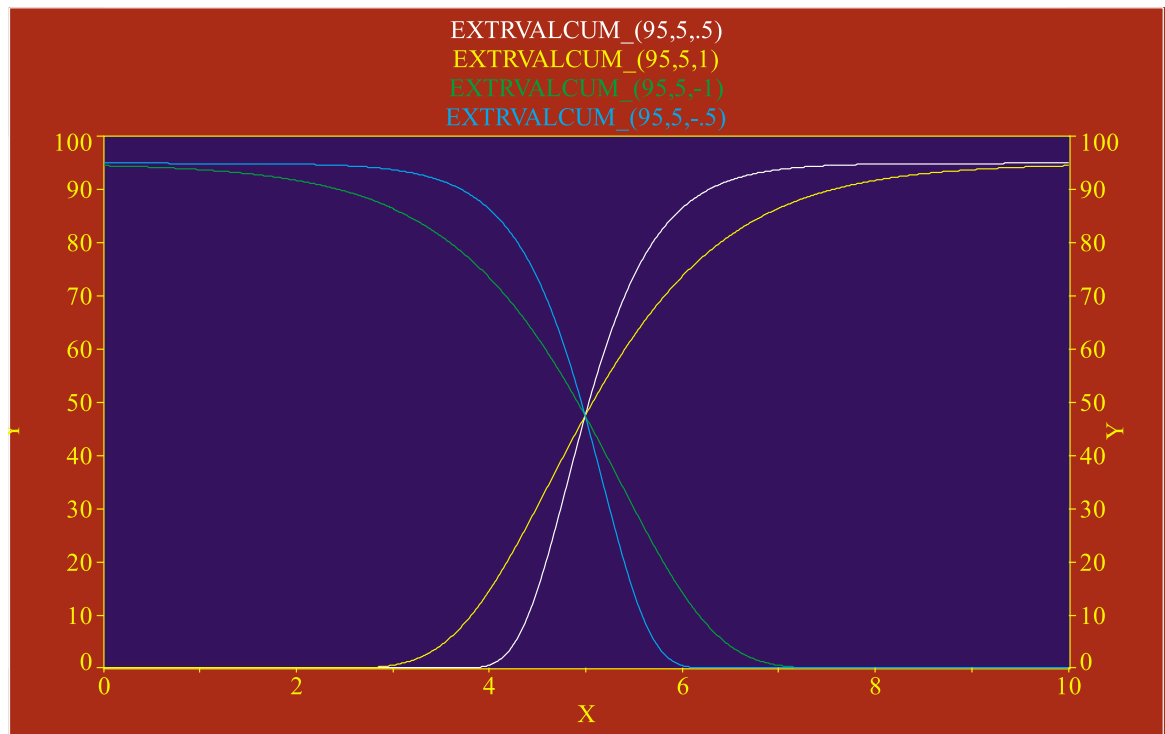
Transition Center: b

Transition Width: $b(\exp(0.6744897500c) - \exp(-0.6744897500c))$

Constraints: $x > 0$, $b > 0$, $c \neq 0$

Built-in Function: **LOGNORMCUM_(a,b,c)**

Extreme Value Cumulative



$$y = a \exp \left(-\exp \left(-\frac{x - c \ln(\ln 2) - b}{c} \right) \right)$$

Non-Intercept Form: Equation **8081**

Intercept Form: Equation **8082**

Transition Height: a

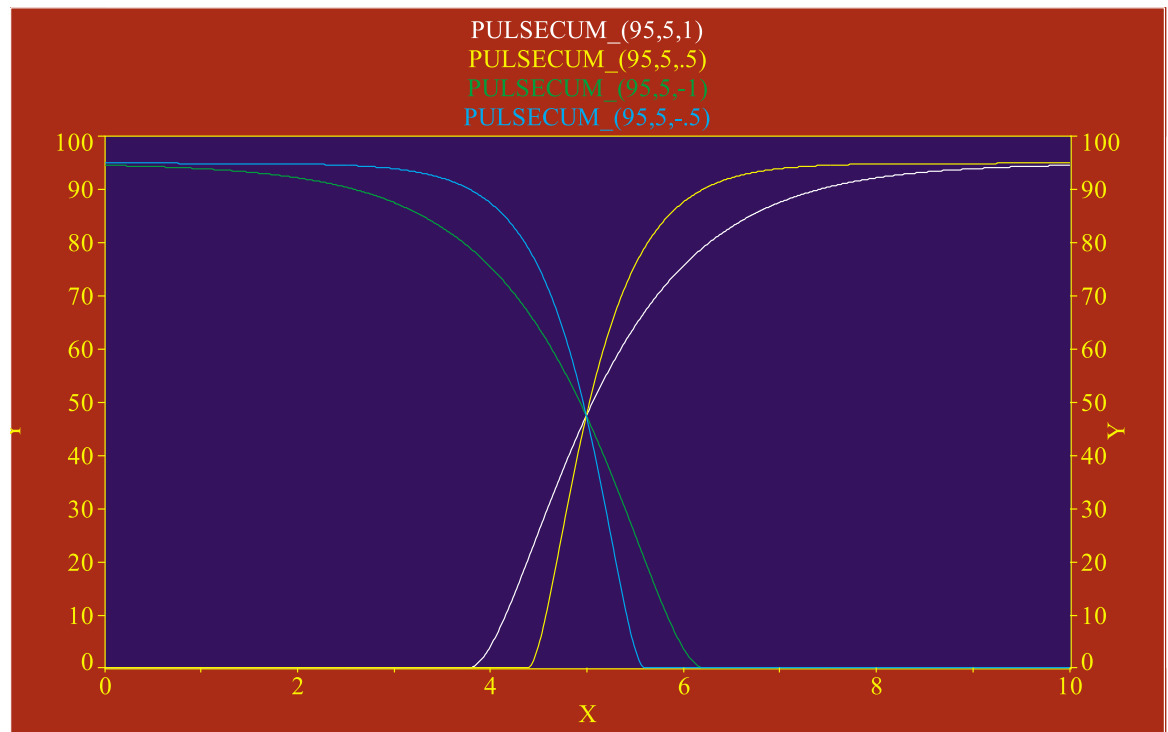
Transition Center: b

Transition Width: $1.572533585c$

Constraints: $c \neq 0$

Built-in Function: **EXTRVALCUM_(a,b,c)**

Pulse Cumulative



$$y = a \left(1 - \exp \left(- \frac{x - c \ln \left(1 - \frac{\sqrt{2}}{2} \right) - b}{c} \right) \right)^2$$

Non-Intercept Form: Equation **8083**

Intercept Form: Equation **8084**

Transition Height: a

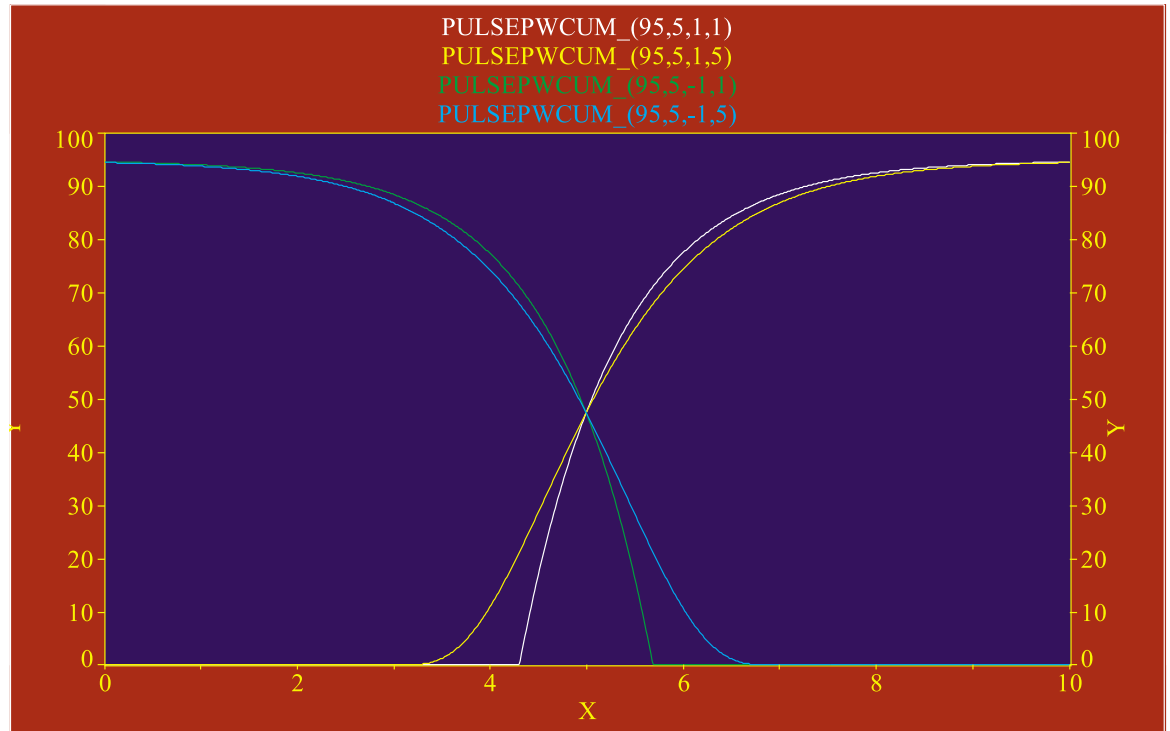
Transition Center: b

Transition Width: $1.3169578969c$

Constraints: $x \geq b + c \ln \left(1 - \frac{\sqrt{2}}{2} \right)$ if $c > 0$, $x \leq b + c \ln \left(1 - \frac{\sqrt{2}}{2} \right)$ if $c < 0$, $c \neq 0$

Built-in Function: **PULSECUM_(a,b,c)**

Pulse Cumulative with Power Term



$$y = a \left[1 - \exp \left(- \frac{x - c \ln \left(1 - 2^{-\frac{1}{d}} \right) - b}{c} \right) \right]^d$$

Non-Intercept Form: Equation **8085**

Intercept Form: Equation **8086**

Transition Height: a

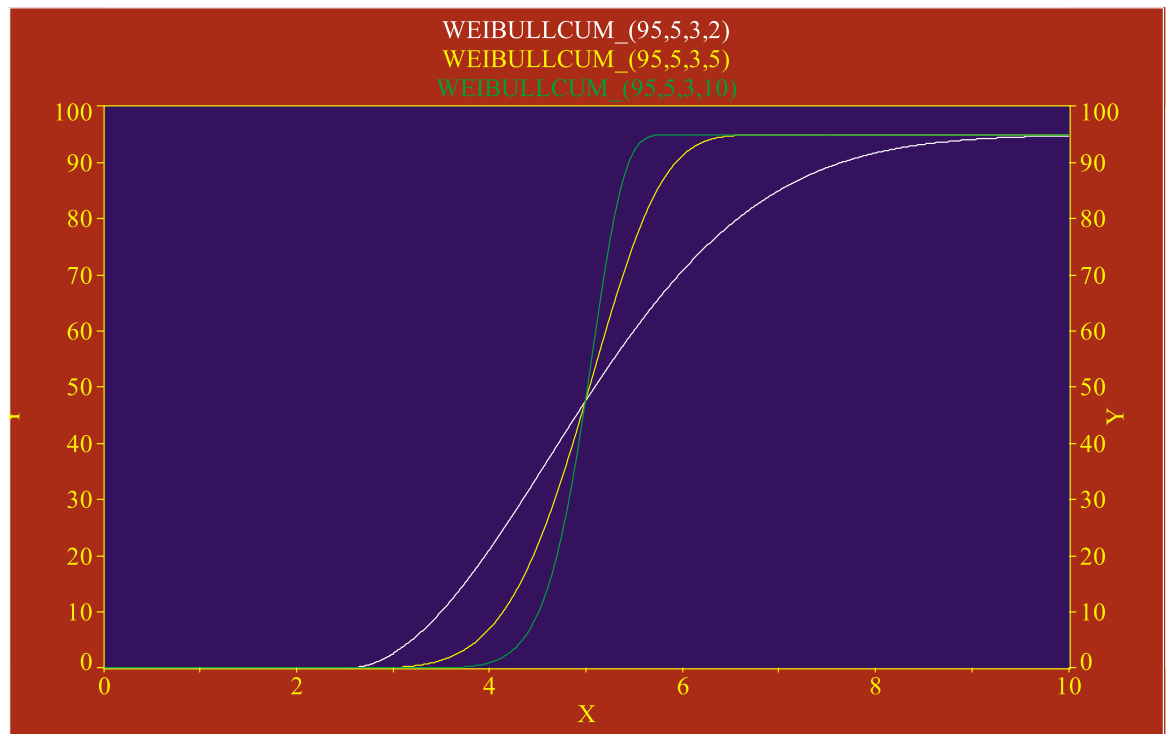
Transition Center: b

Transition Width: Analytic Solution Unknown

Constraints: $x \geq b + c \ln \left(1 - 2^{-1/d} \right)$ if $c > 0$, $x \leq b + c \ln \left(1 - 2^{-1/d} \right)$ if $c < 0$, $c \neq 0$, $d > 0$

Built-in Function: **PULSEPWCUM_(a,b,c,d)**

Weibull Cumulative



$$y = a \left[1 - \exp \left(- \left(\frac{x + c(\ln 2)^{1/d} - b}{c} \right)^d \right) \right]$$

Non-Intercept Form: Equation **8087**

Intercept Form: Equation **8088**

Transition Height: a

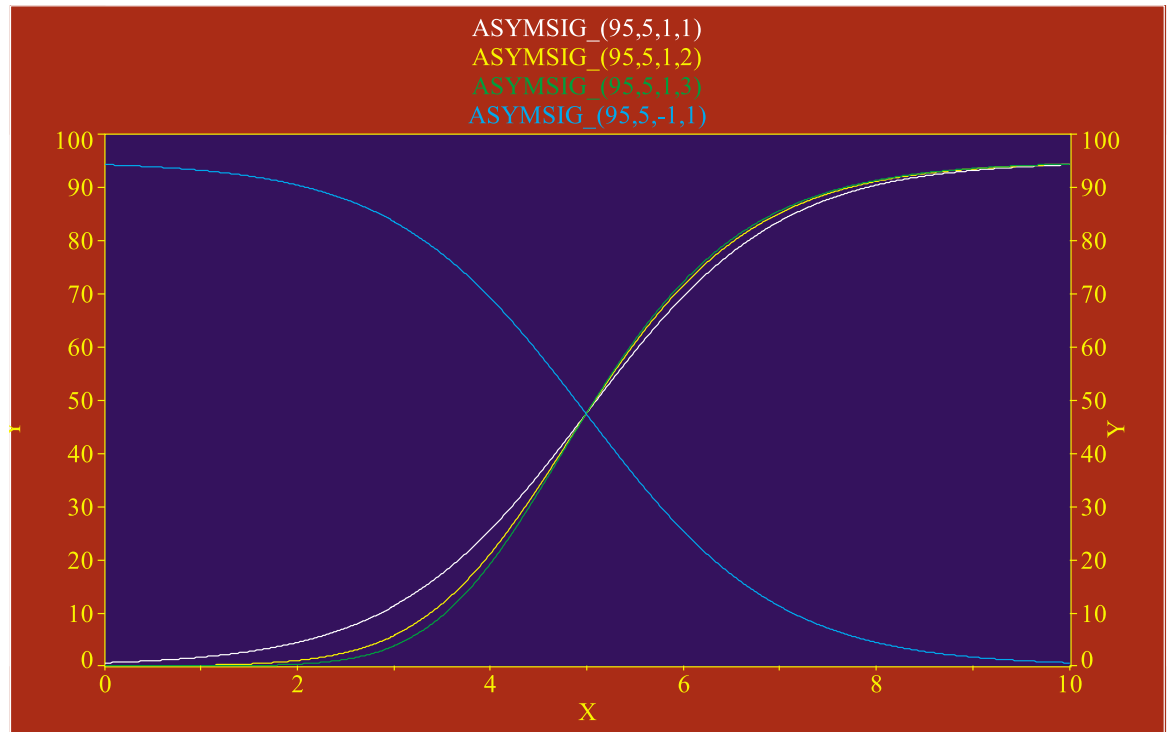
Transition Center: b

Transition Width: $2^{1/d} c (\ln 2)^{1/d} - c (2 \ln 2 - \ln 3)^{1/d}$

Constraints: $c > 0, d > 0, x > b - c(\ln 2)^{1/d}$

Built-in Function: **WEIBULLCUM_(a,b,c,d)**

Asymmetric Sigmoid



$$y = \frac{a}{\left[1 + \exp \left(- \frac{x - c \ln \left(2^{1/d} - 1 \right) - b}{c} \right) \right]^d}$$

Non-Intercept Form: Equation **8089**

Intercept Form: Equation **8090**

Transition Height: a

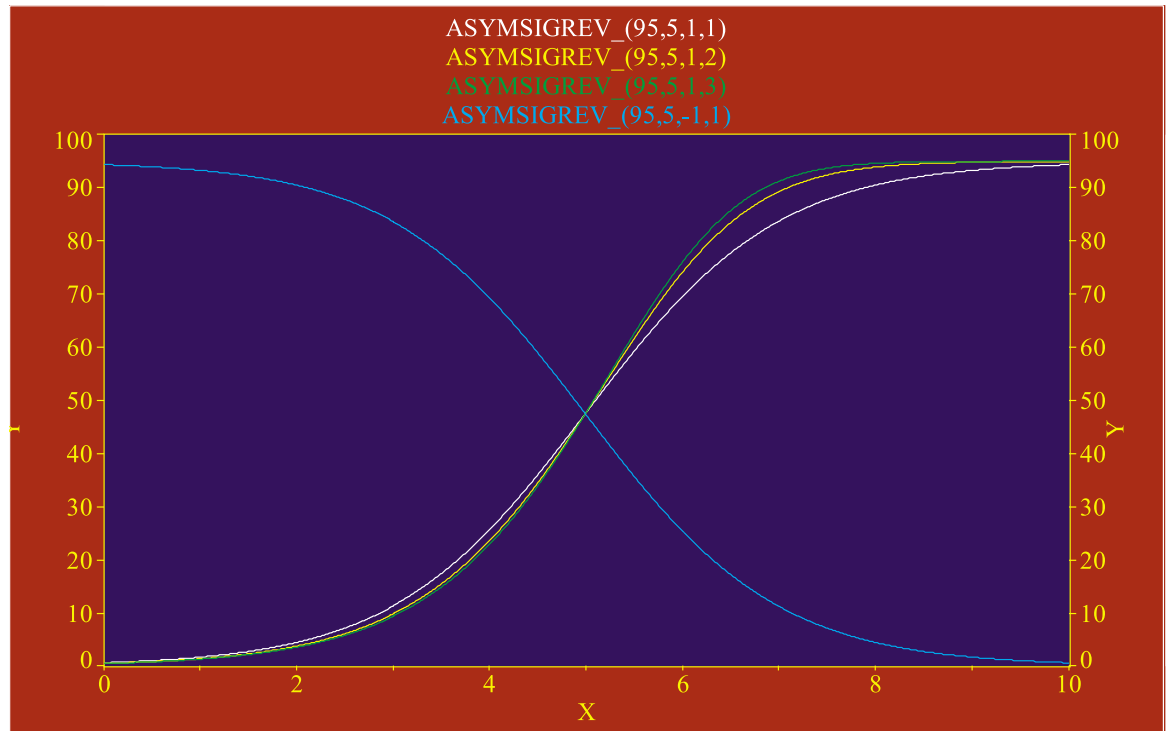
Transition Center: b

Transition Width: $c \ln \left(4^{1/d} - 1 \right) - c \ln \left(4^{1/d} 3^{-1/d} - 1 \right)$

Constraints: $c \neq 0, d > 0$

Built-in Function: **ASYMSIG_(a,b,c,d)**

Asymmetric Sigmoid Reverse Asymmetry



$$y = a \left[1 - \left(1 + \exp \left(\frac{x + c \ln \left(2^{1/d} - 1 \right) - b}{c} \right) \right)^{-d} \right]$$

Non-Intercept Form: Equation **8091**

Intercept Form: Equation **8092**

Transition Height: a

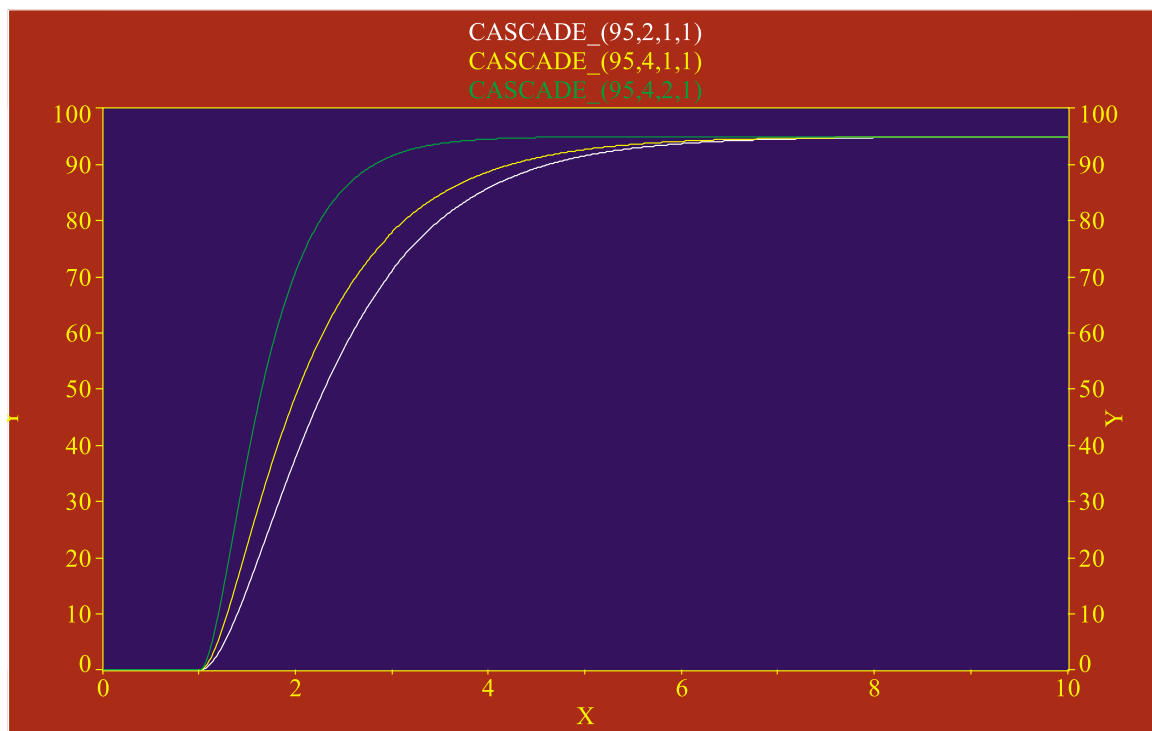
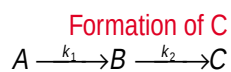
Transition Center: b

Transition Width: $c \ln \left(4^{1/d} - 1 \right) - c \ln \left(4^{1/d} 3^{-1/d} - 1 \right)$

Constraints: $c \neq 0, d > 0$

Built-in Function: **ASYMSIGREV_(a,b,c,d)**

Cacade Transition



$$y = a \left(1 + \frac{b \exp(-c(x-d)) - c \exp(-b(x-d))}{c-b} \right)$$

Non-Intercept Form: Equation **8096**

Intercept Form: Equation **8097**

Max Concentration: a

Rate k_1 : b

Rate k_2 : c

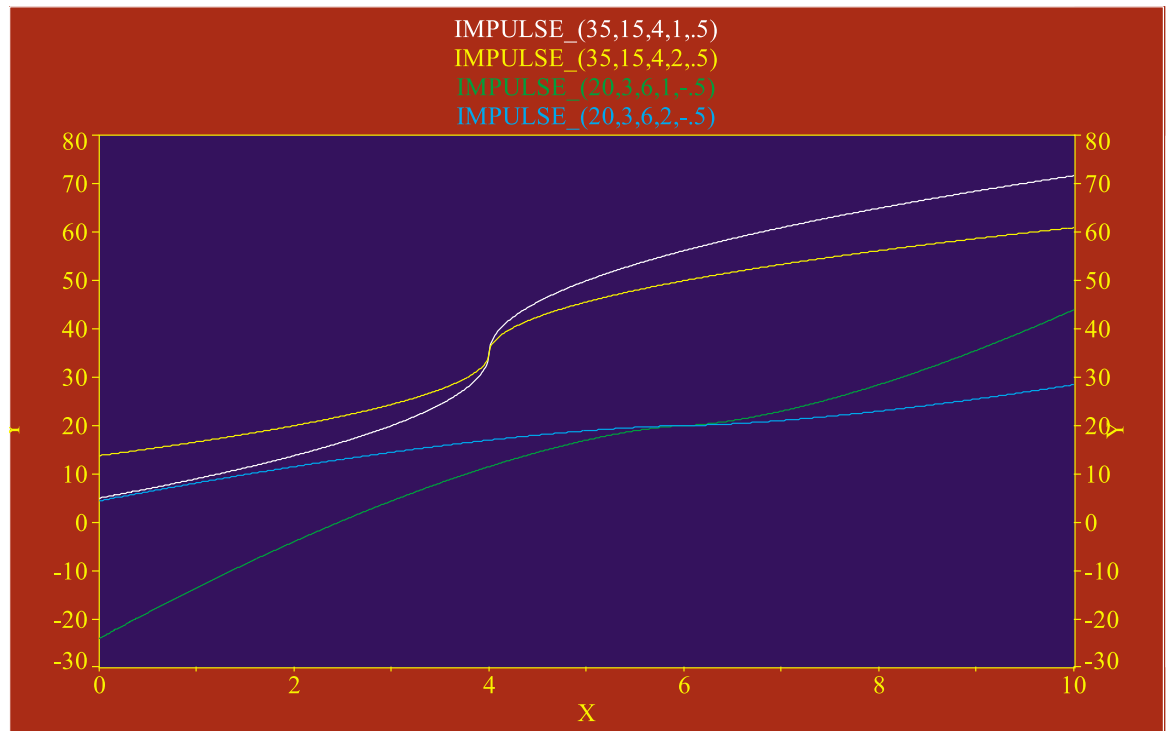
Reaction Begins at time: $x=d$

Constraints: $x>d$, $b>0$, $c>0$, $b \neq c$

Built-in Function: **CASCADe_(a,b,c,d)**

Note that k_1 and k_2 are interchangeable.

Impulse Transition



$$y = a + b \left(\frac{x-c}{d} \right) \left(\frac{|x-c|}{d} \right)^{-e}$$

Equation **8095**

Transition Height: $b/2$

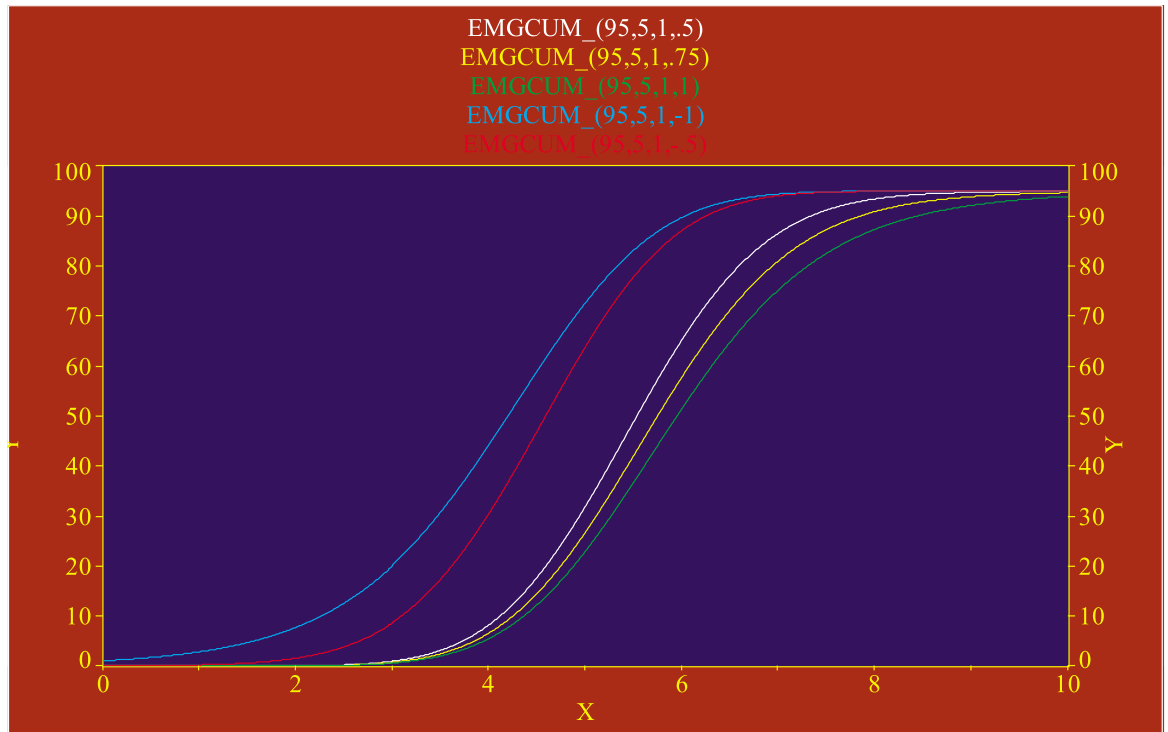
Transition Center x: c

Transition Center y: a

Constraints: $x \neq c$, $d \neq 0$, $-1 \leq e \leq 1$

Built-in Function: **IMPULSE_(a,b,c,d,e)**

Cumulative EMG (Exponentially Modified Gaussian)



$$y = \frac{a}{2} \left[1 + \operatorname{erf} \left(\frac{x-b}{\sqrt{2}c} \right) - \left(\frac{d}{|d|} + \operatorname{erf} \left(\frac{x-b}{\sqrt{2}c} - \frac{c}{\sqrt{2}d} \right) \right) \exp \left(\frac{c^2}{2d^2} + \frac{b-x}{d} \right) \right]$$

Non-Intercept Form: Equation **8188**

Intercept Form: Equation **8189**

Transition Height: a

Transition Center for deconvolved Gaussian cumulative: b

Transition Width for deconvolved Gaussian cumulative: $1.348979501c$

Exponential Time Constant: d

Constraints: $c > 0$, $d \neq 0$

Built-in Function: **EMGCUM_(a,b,c,d)**

Kinetic Equations

- Basic Eqns**
- Half Order Decay and Formation
 - First Order Decay and Formation
 - Second Order Decay and Formation
 - Second Order Decay and Formation, Hyperbolic Forms
 - Third Order Decay and Formation
 - Variable Order Decay and Formation

- Complex Eqns**
- Simultaneous First and Second Order Decay and Formation
 - First Order Sequential Formation
 - Two Component First Order Decay
 - Two First Order Independent Decay and Formation
 - Two Second Order Independent Decay and Formation
 - First and Second Order Independent Decay and Formation

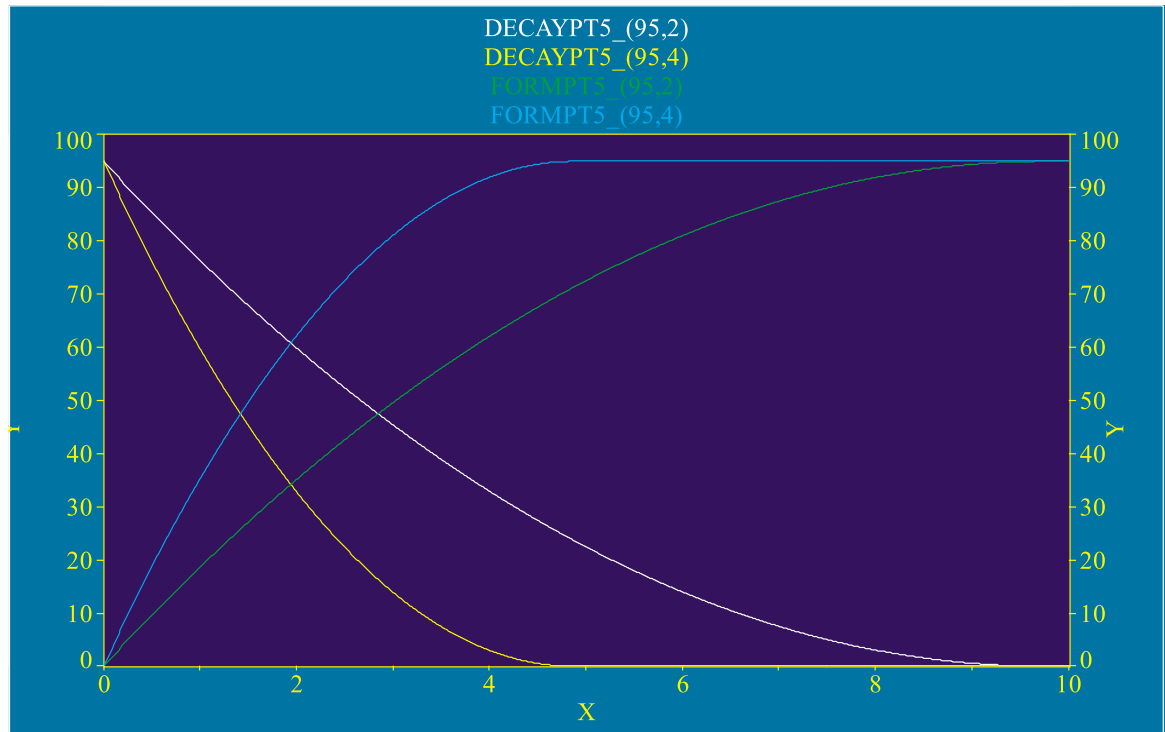
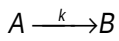
- Equilibrium Eqns**
- Simple Equilibrium, Forward and Reverse Rate
 - Simple Equilibrium, Net Rate and Equilibrium Concentration
 - Complex Equilibrium $A=B+C$
 - Complex Equilibrium $A+B=C+D$

- Intermediate Eqns**
- First Order Intermediate
 - First Order Intermediate with Equilibrium

All reactions begin at time $x=0$.

For the basic equations, a is the maximum concentration, b is the rate constant, and c , if present, is the computed order. x_{50} is the time where the concentration is half the maximum.

Half Order Decay and Formation



Maximum concentration: a

Rate Constant: b

Decay $y = a - \sqrt{abx} + \frac{b^2 x^2}{4}$

Non-Intercept form is Eq. **8114**, intercept form is Eq. **8115**.

Formation $y = \sqrt{abx} - \frac{b^2 x^2}{4}$

Non-Intercept form is Eq. **8116**, intercept form is Eq. **8117**.

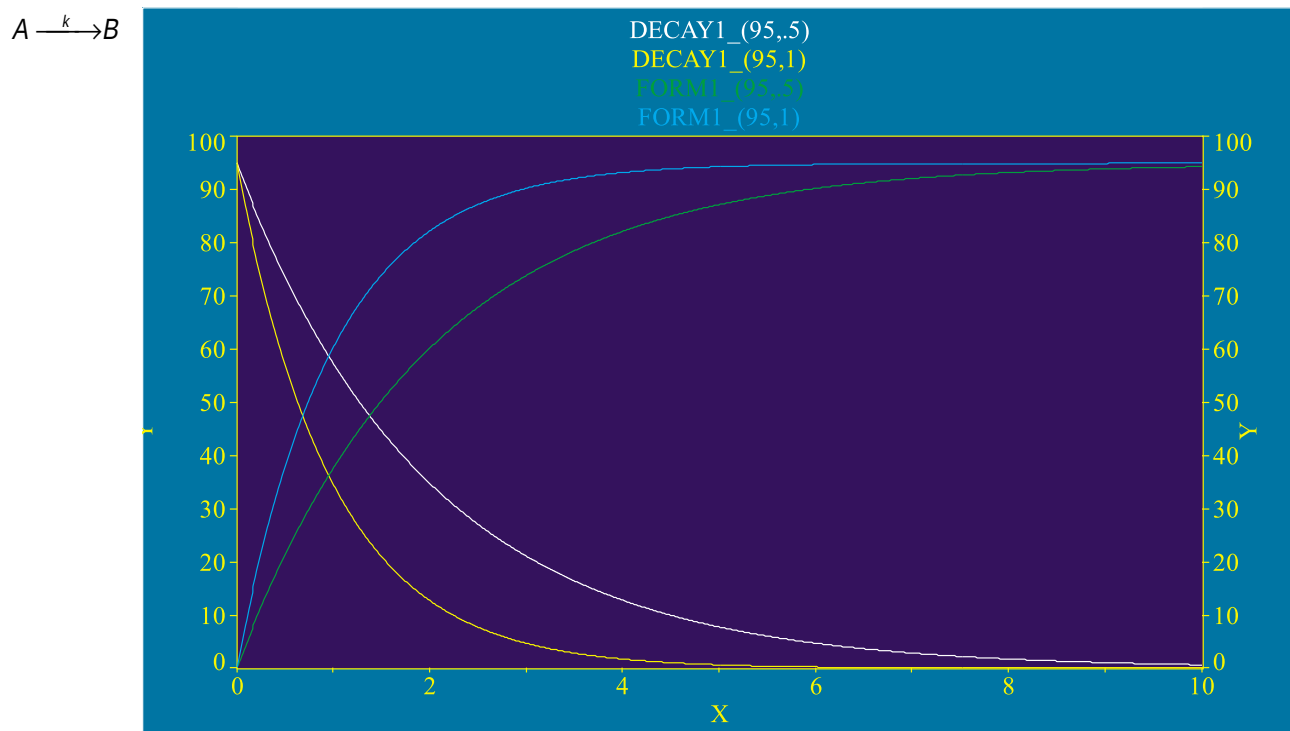
Rate Equation $\frac{dC}{dt} = -k\sqrt{C}$

$x_{50} = \frac{2b\sqrt{a} - b\sqrt{2a}}{b^2}$

Constraints $x \geq 0, a > 0, b > 0, x \leq 2\sqrt{\frac{a}{b}}$

Built-In Functions **DECAYP5_(a,b)** and **FORMPT5_(a,b)**

First Order Decay and Formation



Maximum concentration: a

Rate Constant: b

Decay $y = a \exp(-bx)$

Non-Intercept form is Eq. 8098, intercept form is Eq. 8099.

Formation $y = a(1 - \exp(-bx))$

Non-Intercept form is Eq. 8100, intercept form is Eq. 8101.

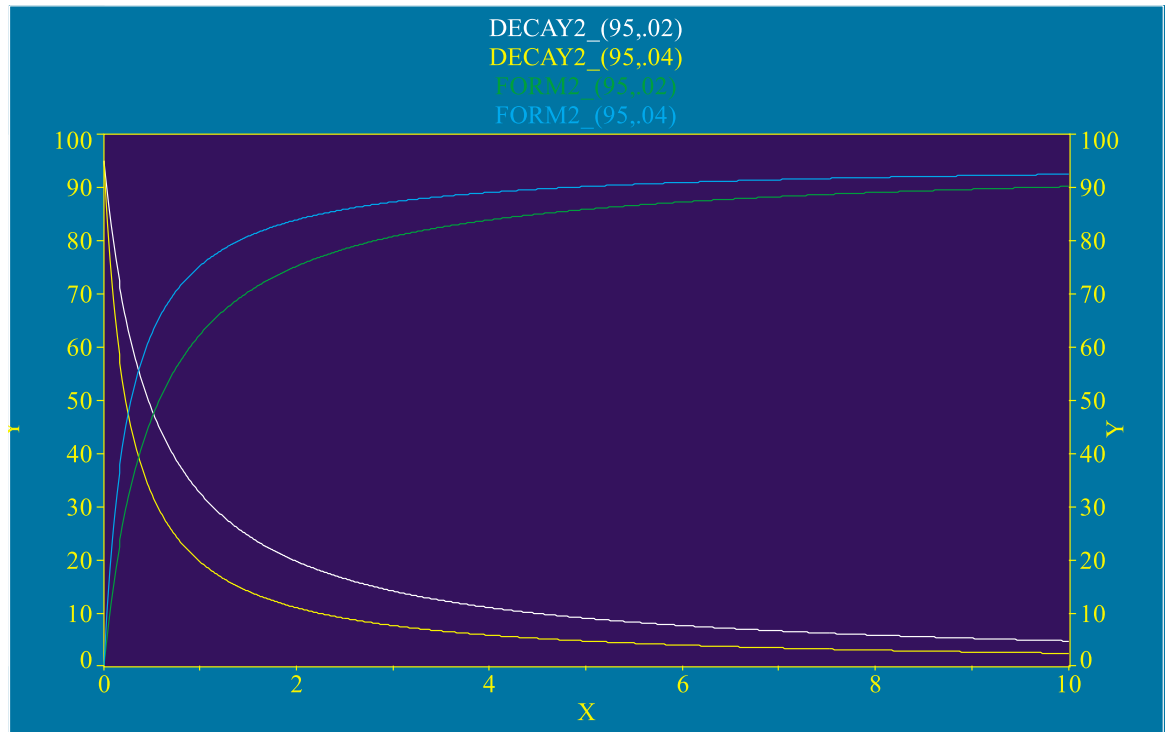
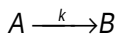
Rate Equation $\frac{dC}{dt} = -kC$

$x_{50} = \ln 2 / b$

Constraints $x \geq 0, a > 0, b > 0$

Built-In Functions **DECA11**(a,b) and **FORM1**(a,b)

Second Order Decay and Formation



Maximum concentration: a

Rate Constant: b

Decay $y = \frac{a}{1 + abx}$

Non-Intercept form is Eq. 8102, intercept form is Eq. 8103.

Formation $y = a \left(1 - \frac{1}{1 + abx} \right)$

Non-Intercept form is Eq. 8104, intercept form is Eq. 8105.

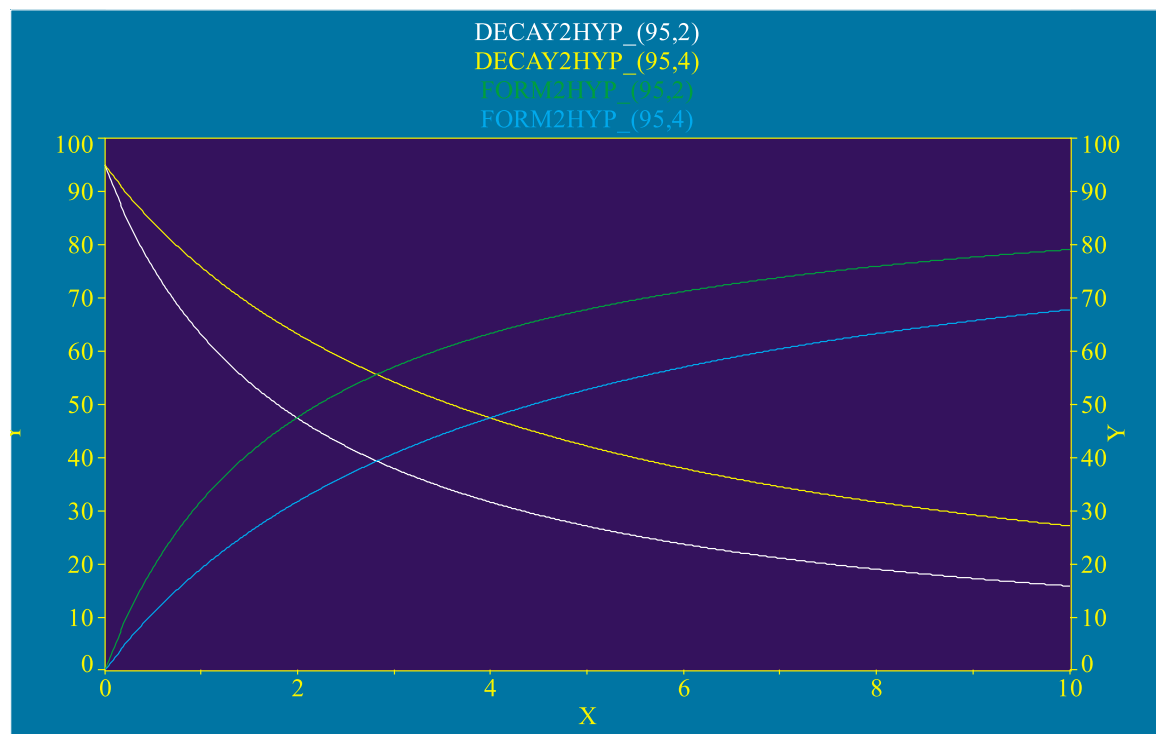
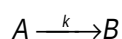
Rate Equation $\frac{dC}{dt} = -kC^2$

$x_{50} = \frac{1}{ab}$

Constraints $x \geq 0, a > 0, b > 0$

Built-In Functions **DECAY2_(a,b)** and **FORM2_(a,b)**

Second Order Decay and Formation, Hyperbolic Forms



Maximum concentration: a

Rate Constant: b

Decay $y = \frac{ab}{b+x}$

Non-Intercept form is Eq. 8106, intercept form is Eq. 8107.

Formation $y = \frac{ax}{b+x}$

Non-Intercept form is Eq. 8108, intercept form is Eq. 8109.

Rate Equation $\frac{dC}{dt} = -\frac{1}{ab}C^2, k = \frac{1}{ab}$

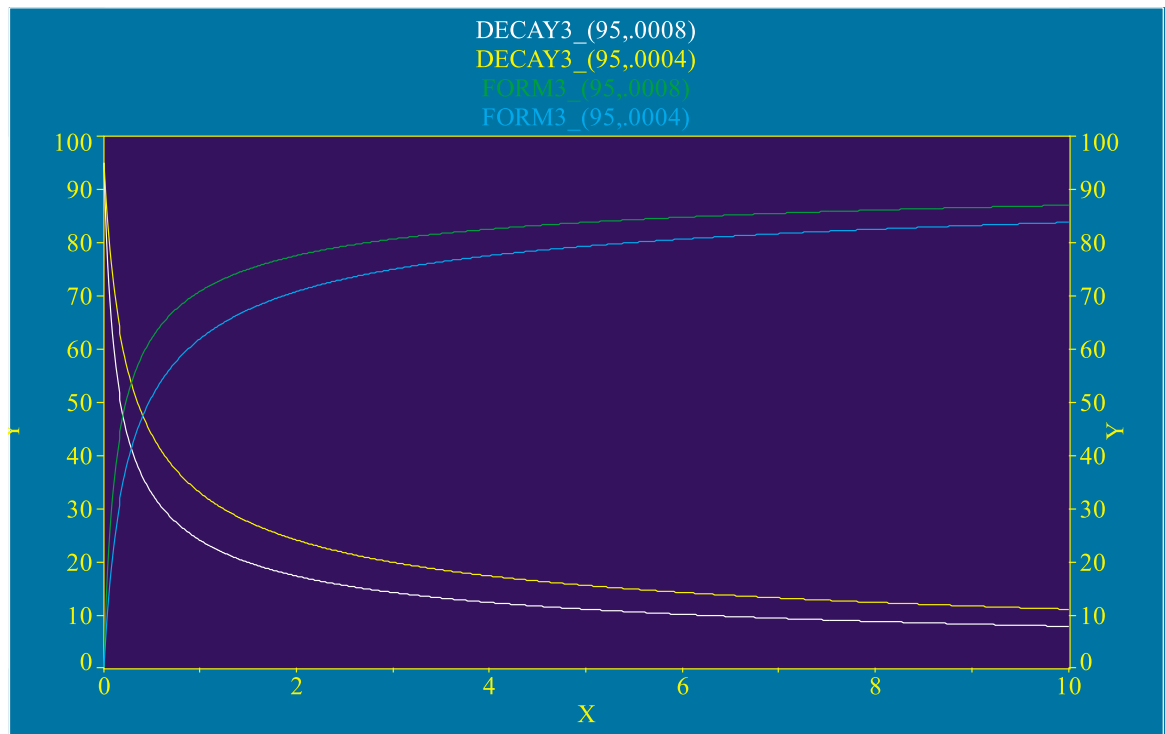
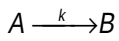
x_{50} b

Constraints $x \geq 0, a > 0, b > 0$

Built-In Functions **DECAY2HYP_(a,b)** and **FORM2HYP_(a,b)**

Second order reactions are often fitted using the simpler hyperbolic forms, since these directly fit the x_{50} . These hyperbolic forms will produce the same goodness of fit statistics as the natural second order forms.

Third Order Decay and Formation



Maximum concentration: a

Rate Constant: b

Decay $y = \frac{a}{\sqrt{1+2a^2 bx}}$

Non-Intercept form is Eq. 8110, intercept form is Eq. 8111.

Formation $y = a \left(1 - \frac{1}{\sqrt{1+2a^2 bx}} \right)$

Non-Intercept form is Eq. 8112, intercept form is Eq. 8113.

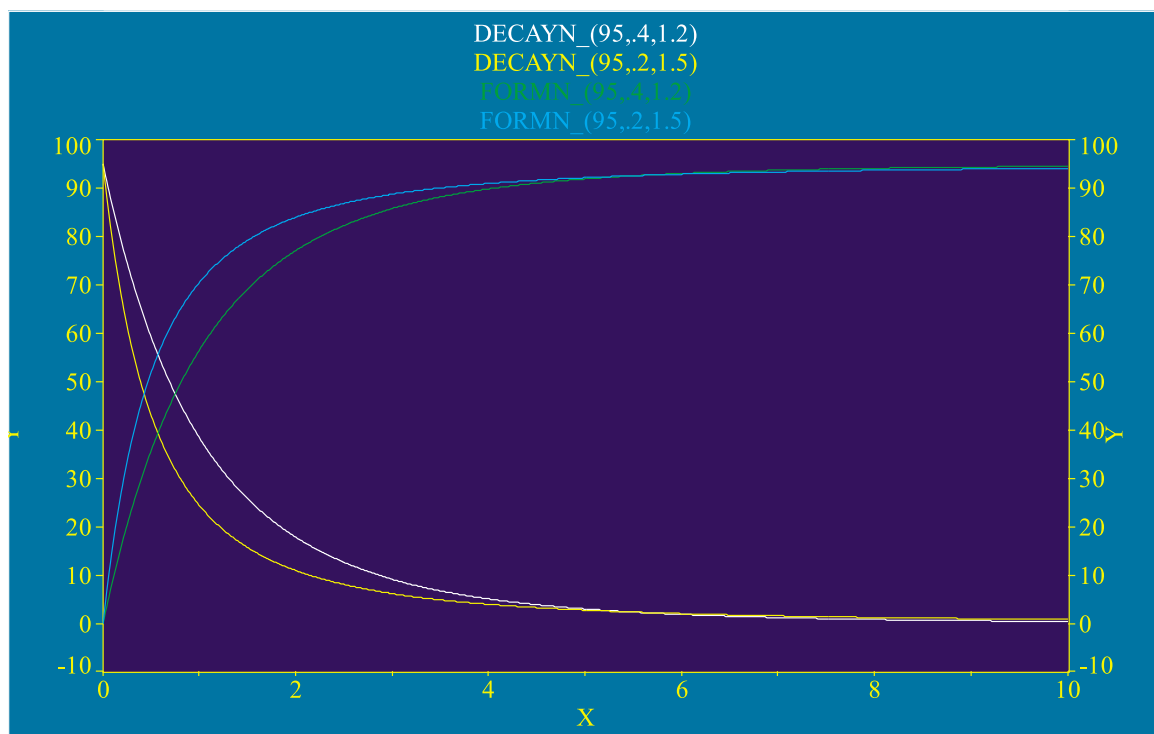
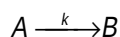
Rate Equation $\frac{dC}{dt} = -kC^3$

$X_{50} \frac{3}{2a^2 b}$

Constraints $x \geq 0, a > 0, b > 0$

Built-In Functions **DECAY3**_ (a,b) and **FORM3**_ (a,b)

Variable Order Decay and Formation



Maximum concentration: a

Rate Constant: b

Order n : c

Decay $y = \left(a^{1-c} + bcx - bx \right)^{1/(1-c)}$

Non-Intercept form is Eq. 8118, intercept form is Eq. 8119.

Formation $y = a - \left(a^{1-c} + bcx - bx \right)^{1/(1-c)}$

Non-Intercept form is Eq. 8120, intercept form is Eq. 8121.

Rate Equation $\frac{dC}{dt} = -kC^n$

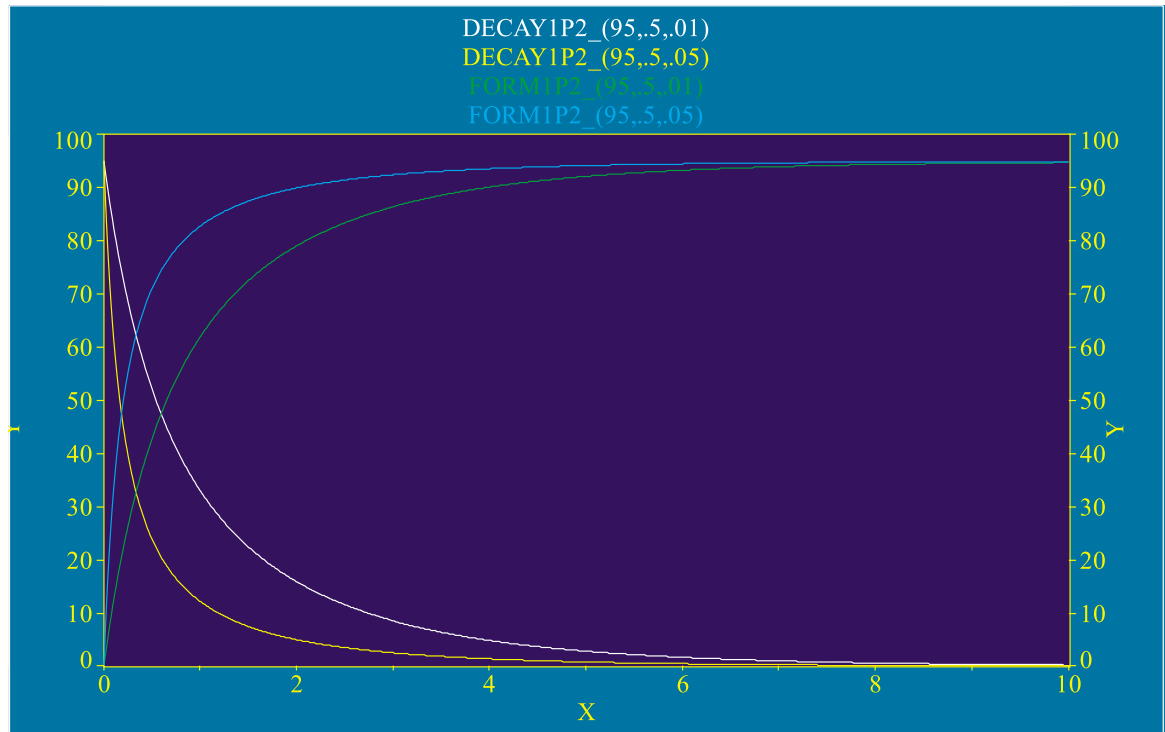
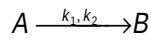
$x_{50} = \frac{\left(\frac{a}{2} \right)^{1-c} - a^{1-c}}{bc - b}$

Constraints $x \geq 0, a > 0, b > 0, c > 0, c \neq 1, x \leq \frac{a^{1-c}}{b(1-c)}$ if $c < 1$

Built-In Functions **DECAYN**_ (a,b,c) and **FORMN**_ (a,b,c)

Note: For accurate determination of the order n , it is recommended that the zero intercept forms of these functions be used. This means that any baseline present should be removed from the data set prior to fitting.

Simultaneous First and Second Order Decay and Formation



Maximum concentration: a

Rate Constants: first order k_1 is b , second order k_2 is c

This model assumes simultaneous first and second order reaction pathways.

Decay $y = \frac{ab}{(ac + b)\exp(bx) - ac}$

Non-Intercept form is Eq. 8122, intercept form is Eq. 8123.

Formation $y = a \left(1 - \frac{b}{(ac + b)\exp(bx) - ac} \right)$

Non-Intercept form is Eq. 8124, intercept form is Eq. 8125.

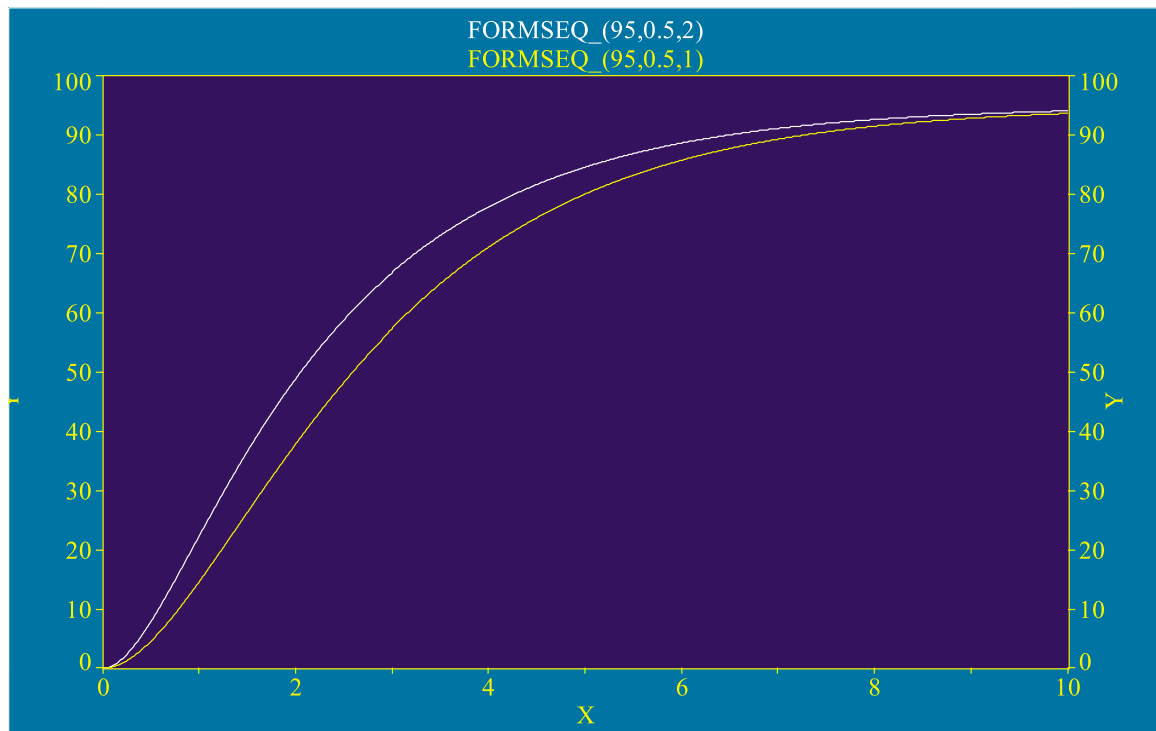
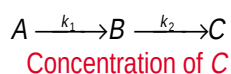
Rate Equation $\frac{dC}{dt} = -k_1 C - k_2 C^2$

$x_{50} \frac{1}{b} \ln \left(\frac{2b + ac}{b + ac} \right)$

Constraints $x \geq 0, a > 0, b > 0, c > 0$

Built-In Functions **DECAY1P2**_ (a,b,c) and **FORM1P2**_ (a,b,c)

First Order Sequential Formation



Maximum concentration: a

First Order Rate Constants: k_1 is b , k_2 is c

$$y = a \left(1 + \frac{b \exp(-cx) - c \exp(-bx)}{c - b} \right)$$

Non-Intercept form is Eq. 8136, intercept form is Eq. 8137

X_{50} Analytic Solution Unknown

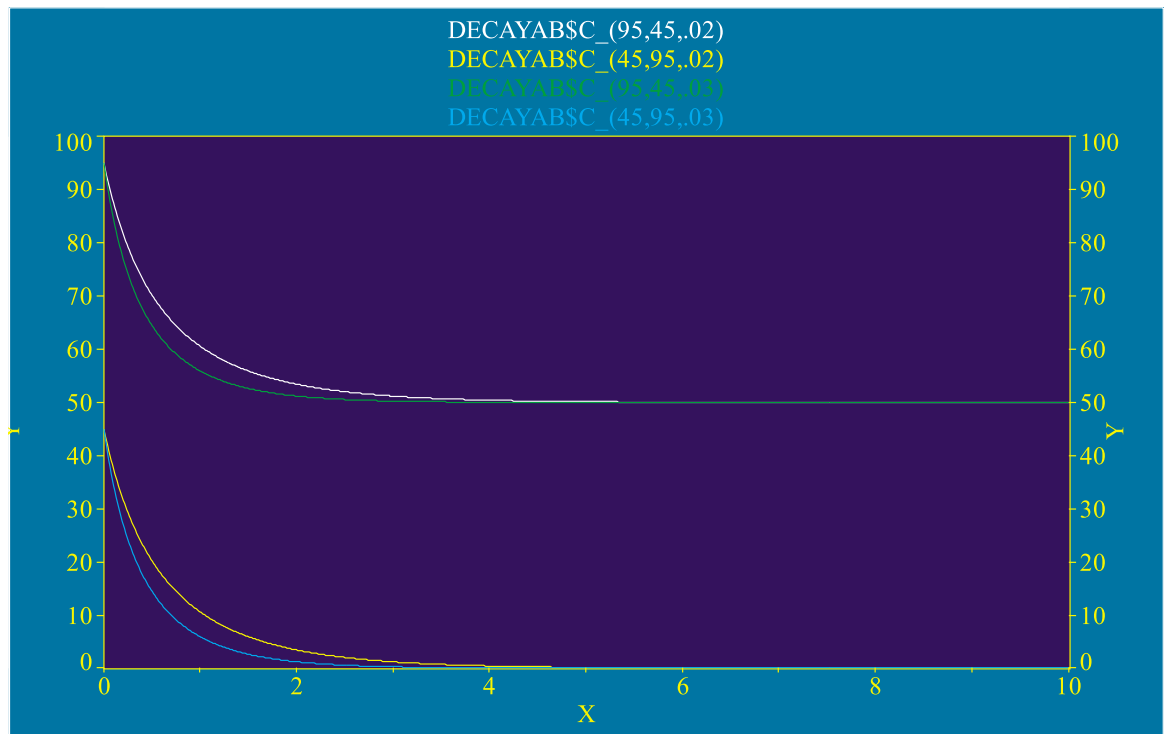
Constraints $x > 0$, $b > 0$, $c > 0$, $b \neq c$

Built-In Function **FORMSEQ_(a,b,c)**

Note that b is the smaller of k_1 and k_2 , and c is the larger (b and c are interchangeable). Note also that the Cascade function within the transition equations is this same sequential formation with an additional term that computes the start of the reaction.

Two Component First Order Decay

$A + B \xrightarrow{k} C$
Concentration of A



Concentration of A at $t=0$: a
Concentration of B at $t=0$: b
Rate Constant k : c

$$y = \frac{a(b-a)}{b \exp(cx(b-a)) - a}$$

Non-Intercept form is Eqs. **8138**, **8140**, intercept form is Eqs. **8139**, **8141**.
Eqs. **8138**, **8139** fit case where $a < b$, Eqs. **8140**, **8141** fit case where $a > b$.

Rate Equation $\frac{dC}{dt} = -k((C_{B_0} - C_{A_0})C + C^2)$

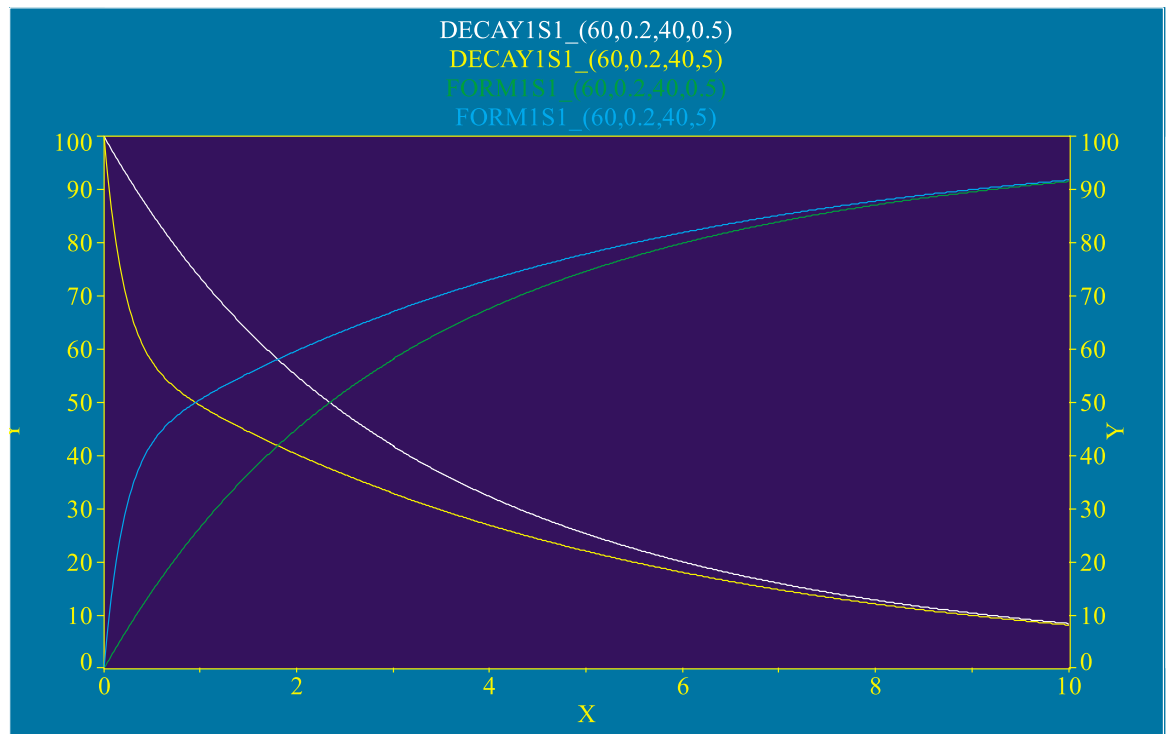
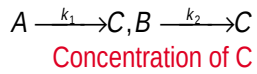
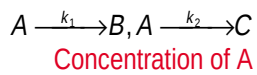
$$X_{50} = \frac{1}{bc - ac} \ln\left(\frac{2b - a}{b}\right)$$

Constraints $x \geq 0$, $a > 0$, $b > 0$, $c > 0$, $a \neq b$

Built-In Functions **DECAyAB\$C_(a,b,c)**

Note that when $a > b$, the final concentration of A represents residual reactant, not an equilibrium.

Two First Order Independent Decay and Formation



Maximum concentration: a

First Order Rate Constants k_1 and k_2 : b and c

Decay $y = a \exp(-bx) + c \exp(-dx)$

Non-Intercept form is Eq. 8144, intercept form is Eq. 8145.

Formation $y = a(1 - \exp(-bx)) + c(1 - \exp(-dx))$

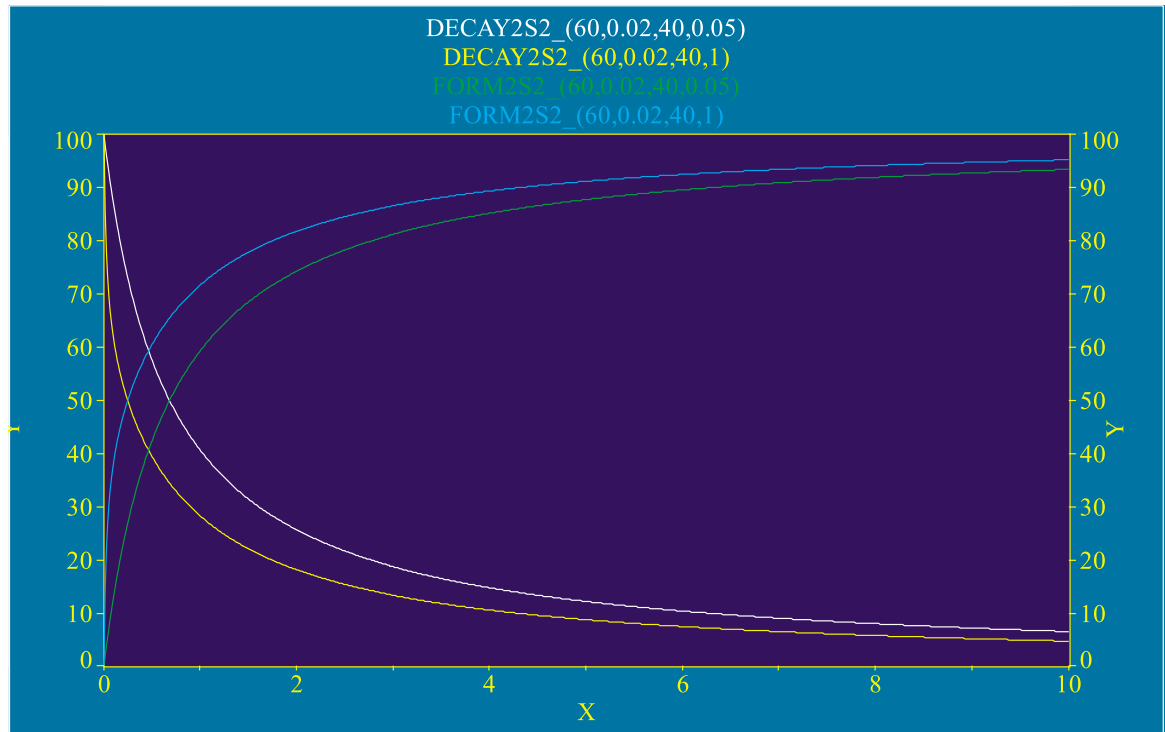
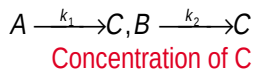
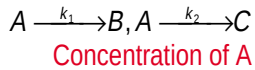
Non-Intercept form is Eq. 8146, intercept form is Eq. 8147.

X_{50} Analytic Solution Unknown

Constraints $x \geq 0, a > 0, b > 0, c > 0, d > 0$

Built-In Functions **DECA1S1_**(a,b,c,d) and **FORM1S1**(a,b,c,d)

Two Second Order Independent Decay and Formation



Maximum concentration: a

Second Order Rate Constants k_1 and k_2 : b and c

Decay
$$y = \frac{a}{1 + abx} + \frac{c}{1 + cdx}$$

Non-Intercept form is Eq. 8152, intercept form is Eq. 8153.

Formation
$$y = a \left(1 - \frac{1}{1 + abx} \right) + c \left(1 - \frac{1}{1 + cdx} \right)$$

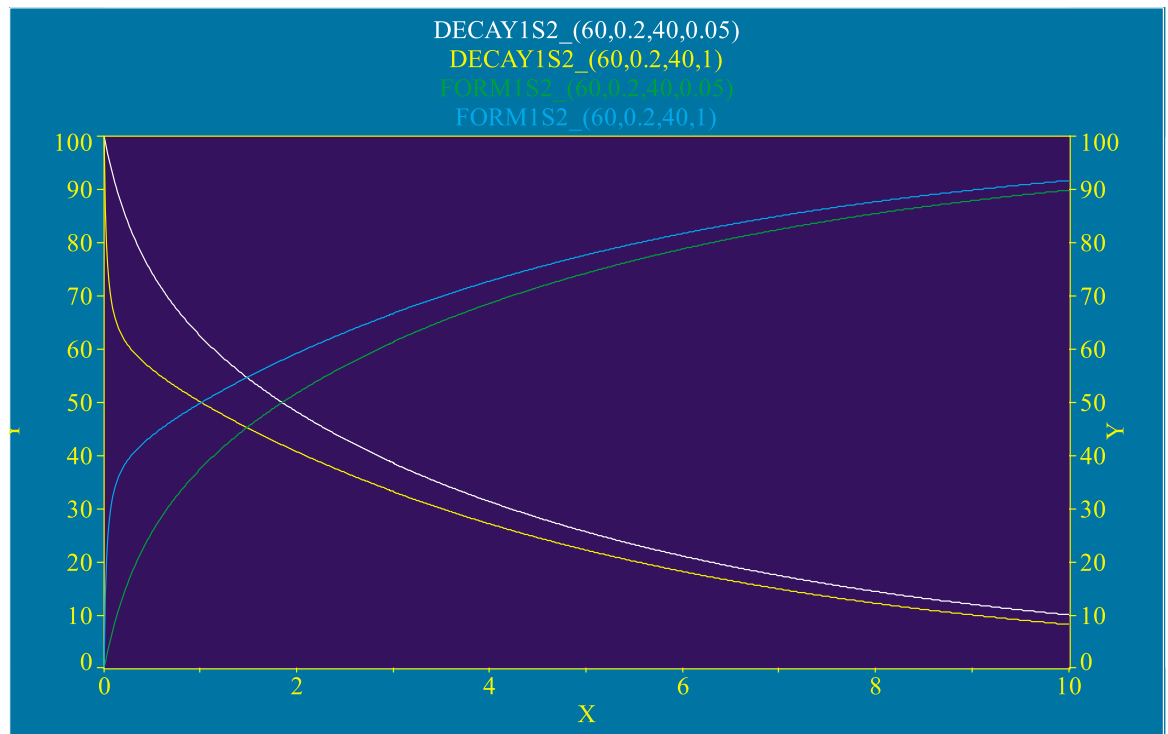
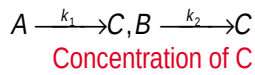
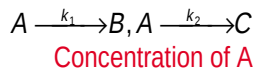
Non-Intercept form is Eq. 8154, intercept form is Eq. 8155.

X_{50} Analytic Solution Unknown

Constraints $x \geq 0, a > 0, b > 0, c > 0, d > 0$

Built-In Functions **DECAY2S2_**(a,b,c,d) and **FORM2S2**(a,b,c,d)

First and Second Order Independent Decay and Formation



Maximum concentration: a
 First Order Rate Constant k_1 : b
 Second Order Rate Constant k_2 : c

Decay $y = a \exp(-bx) + \frac{c}{1 + cdx}$

Non-Intercept form is Eq. 8148, intercept form is Eq. 8149.

Formation $y = a(1 - \exp(-bx)) + c\left(1 - \frac{1}{1 + cdx}\right)$

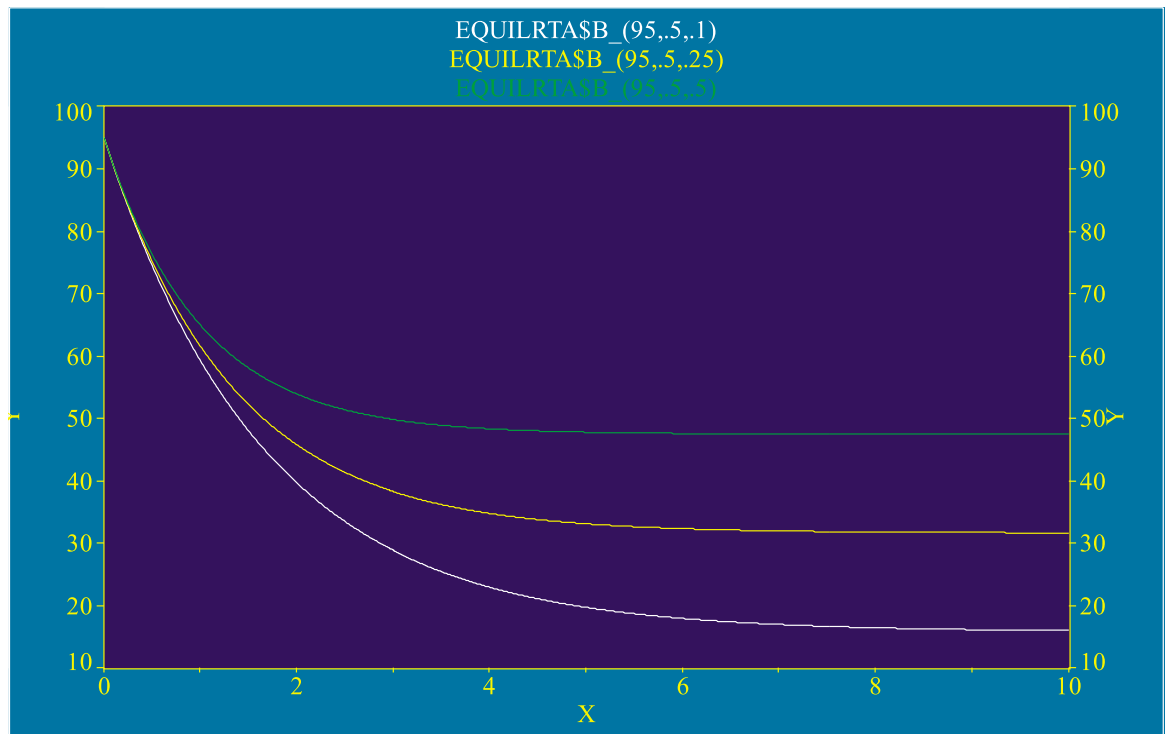
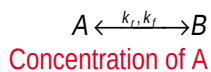
Non-Intercept form is Eq. 8150, intercept form is Eq. 8151.

X_{50} Analytic Solution Unknown

Constraints $x \geq 0, a > 0, b > 0, c > 0, d > 0$

Built-In Functions **DECAY1S2**_(a,b,c,d) and **FORM1S2**(a,b,c,d)

Simple Equilibrium, Forward and Reverse Rate



Maximum concentration: a

Rate Constants k_f and k_r : b and c

$$y = \frac{a(c + b \exp(-(b+c)x))}{b+c} \quad \text{Equation 8142}$$

Rate Equation $\frac{dC}{dt} = -k_f C + k_r (C_{A_0} - C)$

$x_{50} \frac{\ln 2}{(b+c)}$

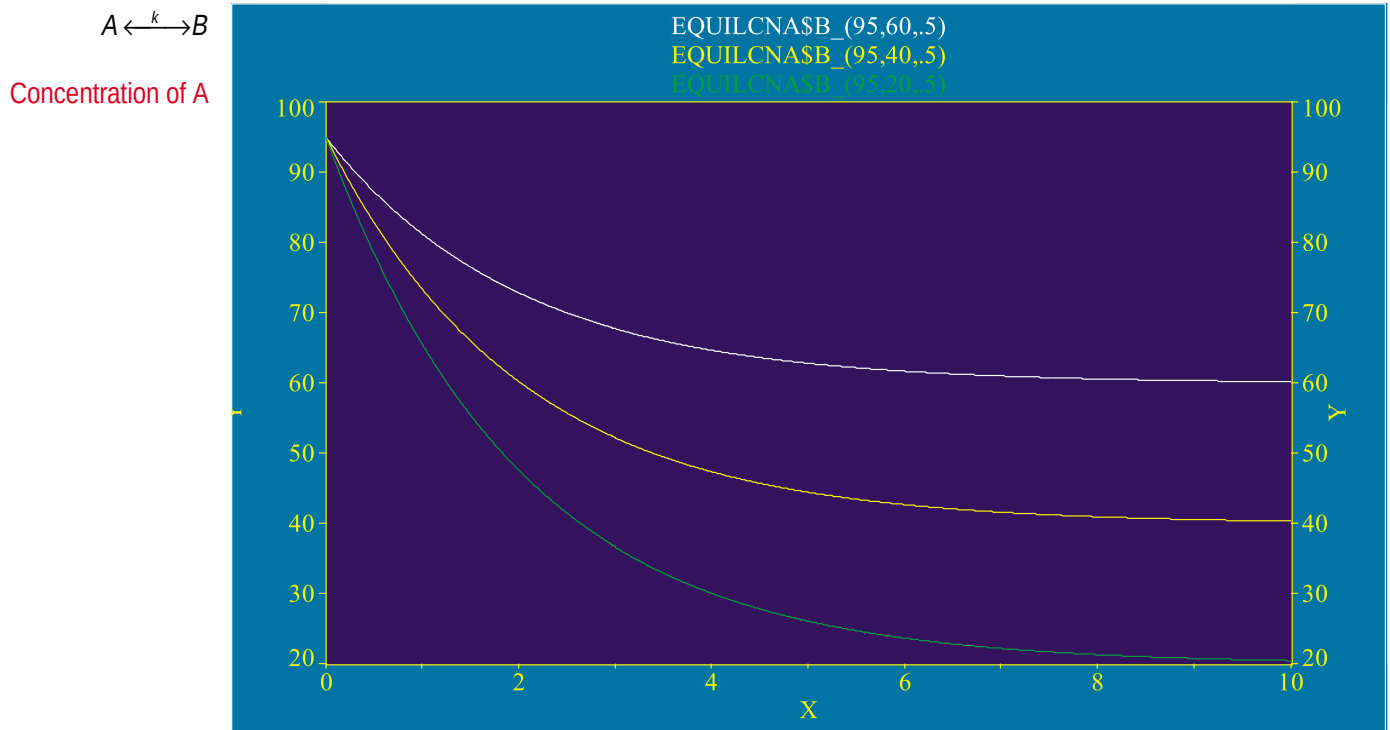
Equilibrium Conc $\frac{ac}{(b+c)}$

Constraints $x \geq 0, b > 0, c > 0$

Built-In Function **EQUILRTASB(a, b, c)**

The reaction is first order in both directions. The initial concentration of B is 0. For this model, x_{50} is the time where the concentration is halfway between a and the equilibrium concentration. Note that Equations 8142 and 8143 produce the same goodness of fit statistics.

Simple Equilibrium, Net Rate and Equilibrium Concentration



Maximum concentration: a
 Equilibrium concentration: b
 Net Rate Constant k : c

$$y = b + (a - b) \exp(-cx) \quad \text{Equation 8143}$$

Rate Equation $\frac{dC}{dt} = -(k_f - k_r)(C - C_{A_E})$

$$x_{50} = \frac{\ln 2}{c}$$

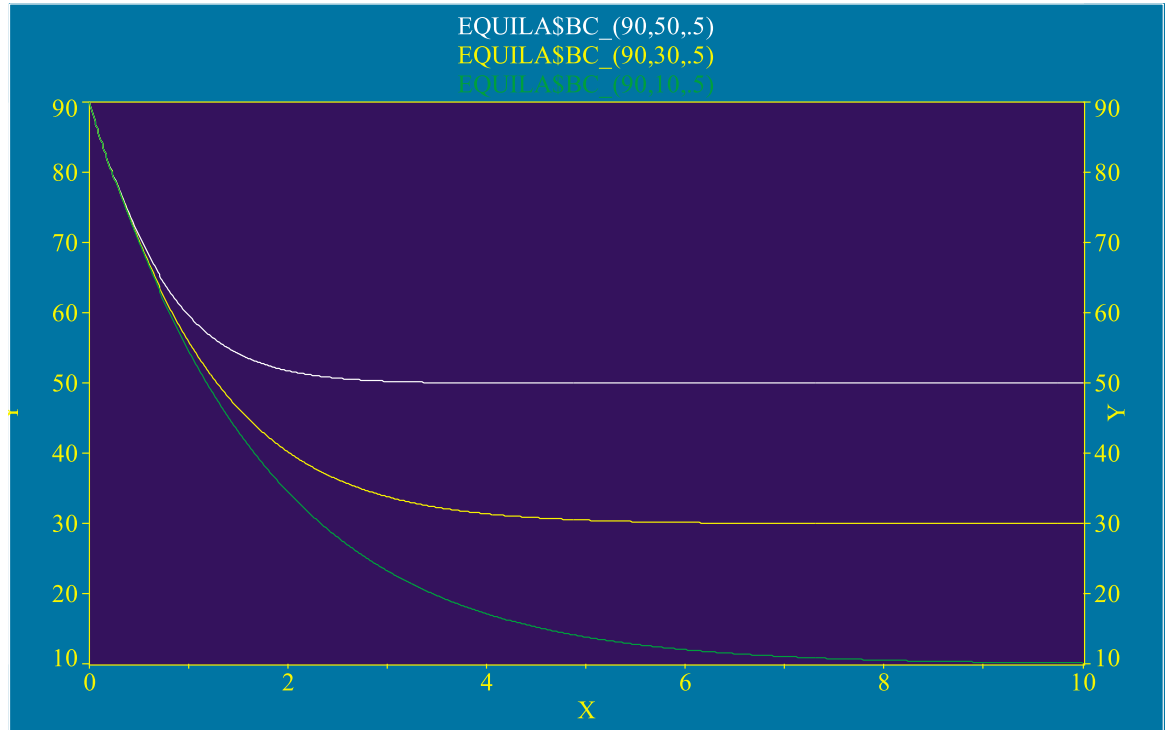
Constraints $x \geq 0, b > 0, c > 0$

Built-In Function **EQUILCNAB(a, b, c)**

The reaction is first order in both directions. The initial concentration of B is 0. For this model, x_{50} is the time where the concentration is halfway between a and b . Note that Equations 8142 and 8143 produce the same goodness of fit statistics.

Complex Equilibrium $A \rightleftharpoons B + C$

$A \xrightleftharpoons{k} B + C$
Concentration of A



Maximum concentration: a
Equilibrium concentration: b
Net Rate Constant k : c

$$y = \frac{a \left[a + b \exp \left(\frac{c(a+b)x}{a-b} \right) \right]}{b + a \exp \left(\frac{c(a+b)x}{a-b} \right)} \quad \text{Equation 8126}$$

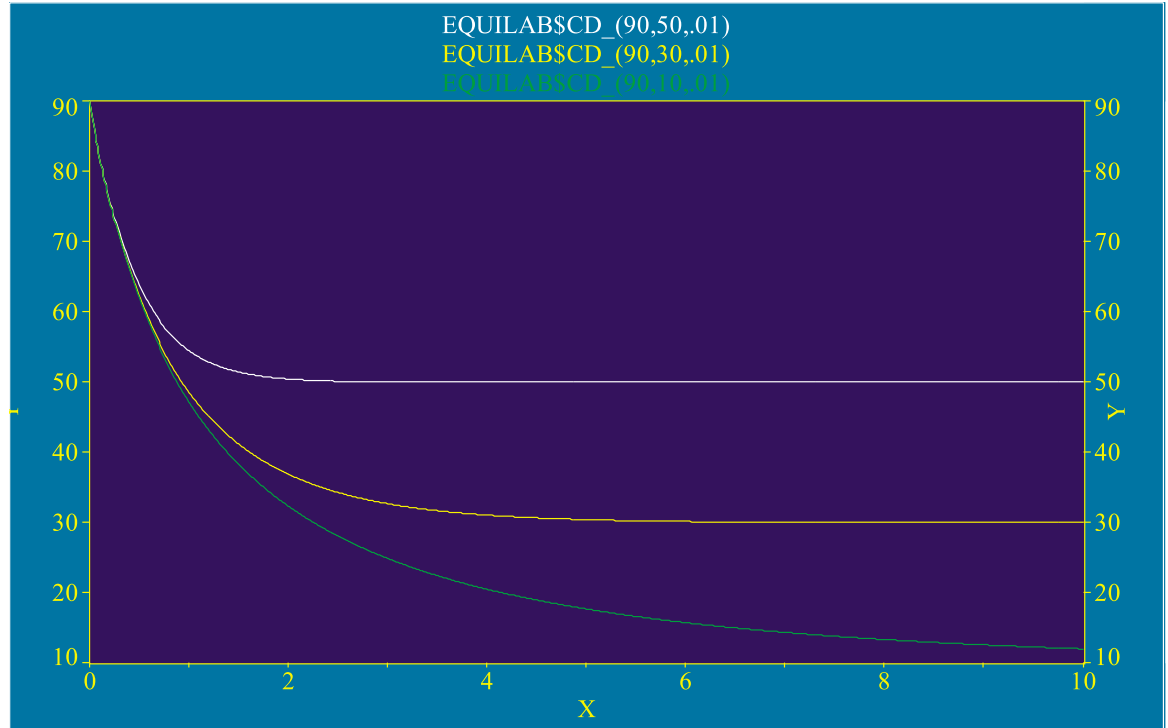
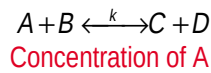
Rate Equation $\frac{dC}{dt} = -(k_f - k_r) \frac{(C_{A_0}^2 - C_{A_E} C)(C_{A_E} - C)}{(C_{A_0} - C_{A_E})^2}$

$x_{50} = \frac{(a-b)(\ln(2a+b) - \ln a)}{c(a+b)}$

Constraints $x \geq 0, a > 0, b > 0, c > 0, a \neq b$

Built-In Function **EQUILA\$BC(a,b,c)**

The reaction is first order in both directions. The initial concentration of B and C is 0. For this model, x_{50} is the time where the concentration is halfway between a and b .

Complex Equilibrium $A+B=C+D$ 

Maximum concentration: a
 Equilibrium concentration: b
 Net Rate Constant k : c

$$y = \frac{ab \left[\exp\left(\frac{2abcx}{a-b}\right) + 1 \right]}{2b - a + a \exp\left(\frac{2abcx}{a-b}\right)} \quad \text{Equation 8127}$$

Rate Equation $\frac{dC}{dt} = -(k_f - k_r) \frac{C_{A_0} (C_{A_0} C_{A_E} + C_{A_0} C - 2C_{A_E} C)(C_{A_E} - C)}{(C_{A_0} - C_{A_E})^2}$

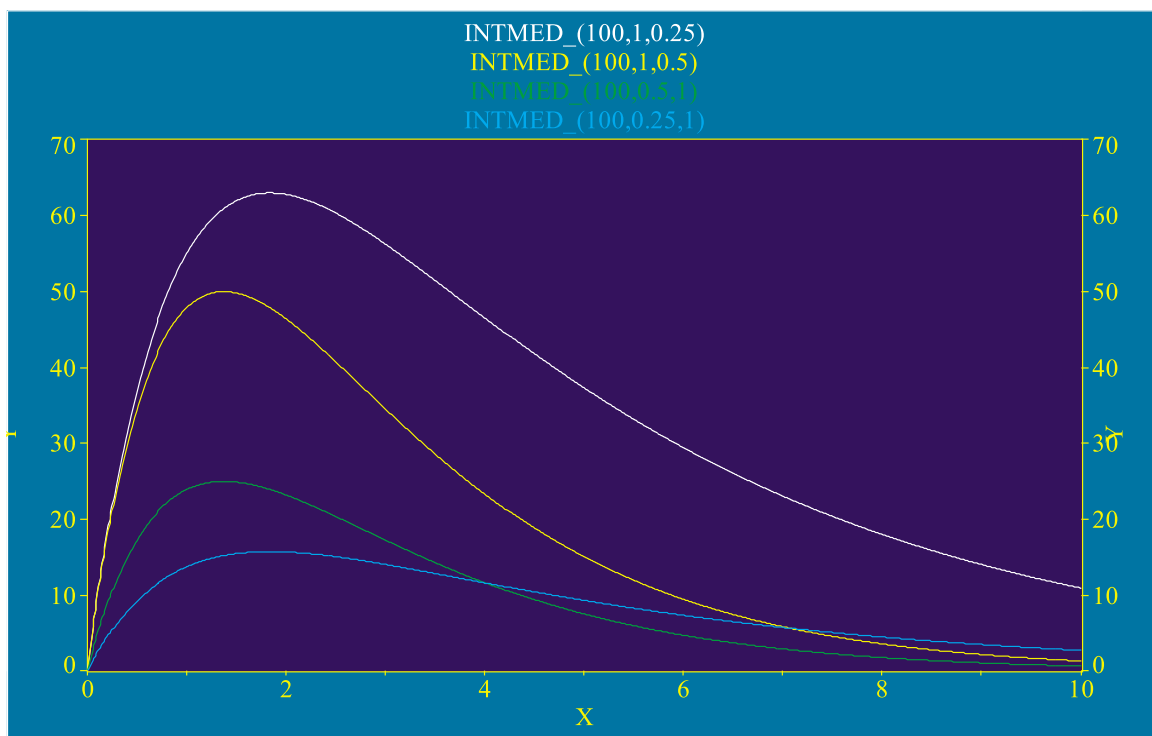
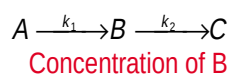
$$x_{50} = \frac{(a-b) \ln\left(\frac{ab + a^2 - 2b^2}{a^2 - ab}\right)}{2abc}$$

Constraints $x \geq 0, a > 0, b > 0, c > 0, a \neq b$

Built-In Function **EQUILAB\$CD(a,b,c)**

The reaction is first order in both directions. The initial concentration of C and D is 0. The initial concentrations of A and B are equal. For this model, x_{50} is the time where the concentration is halfway between a and b .

First Order Intermediate



Initial concentration of A: a
 Rate constants k_1 and k_2 : b and c

$$y = \frac{ab(\exp(-bx) - \exp(-cx))}{c - b}$$

Non-Intercept form: Eqs. **8128**, **8130**, intercept form: Eqs. **8129**, **8131**.
 Eqs. **8128**, **8129** fit case where $b > c$, Eqs. **8130**, **8131** fit case where $b < c$.

Amplitude
Max Conc of B

$$\frac{a - b}{c - b} \left[\exp\left(\frac{b \ln b - b \ln c}{c - b}\right) - \exp\left(\frac{c \ln b - c \ln c}{c - b}\right) \right]$$

X at Max:

$$\frac{(\ln c - \ln b)}{(c - b)}$$

Area

$$\frac{a}{c}$$

Constraints

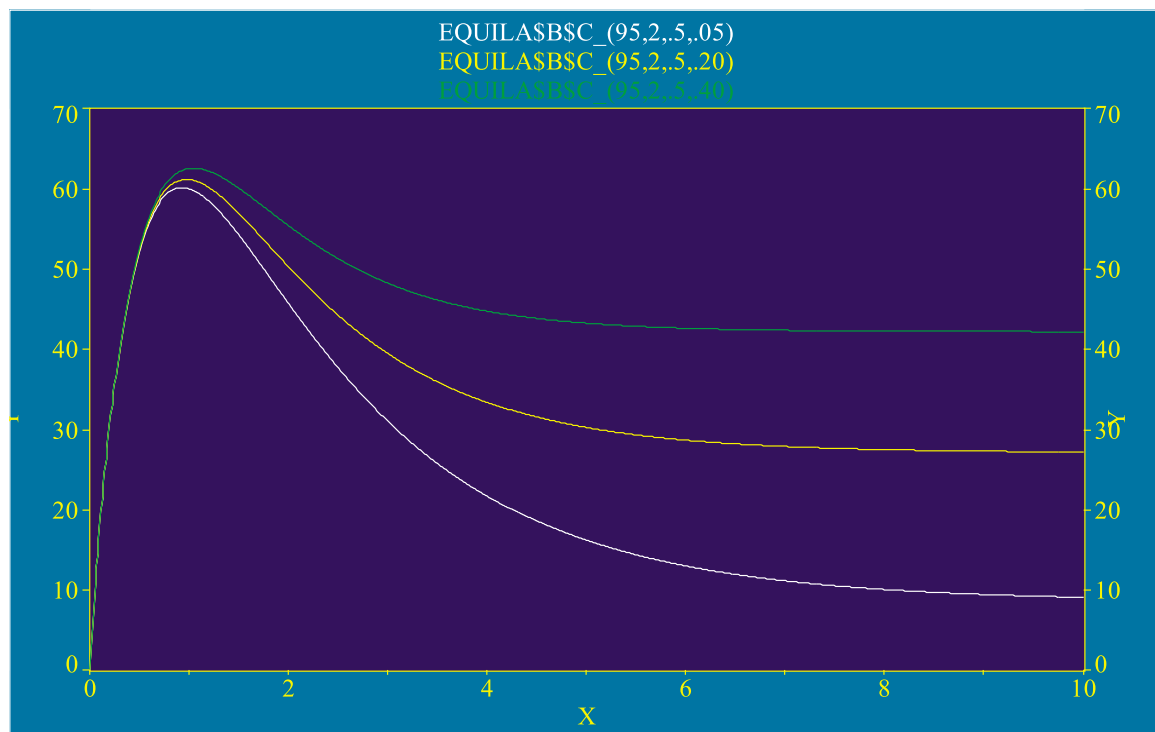
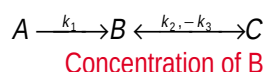
$$x > 0, b > 0, c > 0, b \neq c$$

Built-in Function

$$\text{INTMED\_}(a,b,c)$$

Both steps are first order. These equations are identical to the Intermediate Peak Eqns. 8068 and 8069 in the peak group except that here the reaction begins at time $x=0$.

First Order Intermediate with Equilibrium



Initial concentration of A: a

Rate constants k_1, k_2 , and k_3 : b, c and d

$$y = a \left[1 - \exp(-bx) - \frac{c}{c+d} \left(1 + \frac{b \exp(-(c+d)x) - (c+d) \exp(-bx)}{c+d-b} \right) \right]$$

Non-Intercept form is Eqs. **8132, 8134**; intercept form is Eqs. **8133, 8135**.

Eqs. **8132, 8133** fit case where $b > c$, Eqs. **8134, 8135** fit case where $b < c$.

X at Max $x_c = \frac{\ln(bc(-c-d)) - \ln(b(c+d)(d-b))}{c+d-b}$

Amplitude
Max Conc of B $a \left[1 - \exp(-bx_c) - \frac{c}{c+d} \left(1 + \frac{b \exp(-(c+d)x_c) - (c+d) \exp(-bx_c)}{c+d-b} \right) \right]$

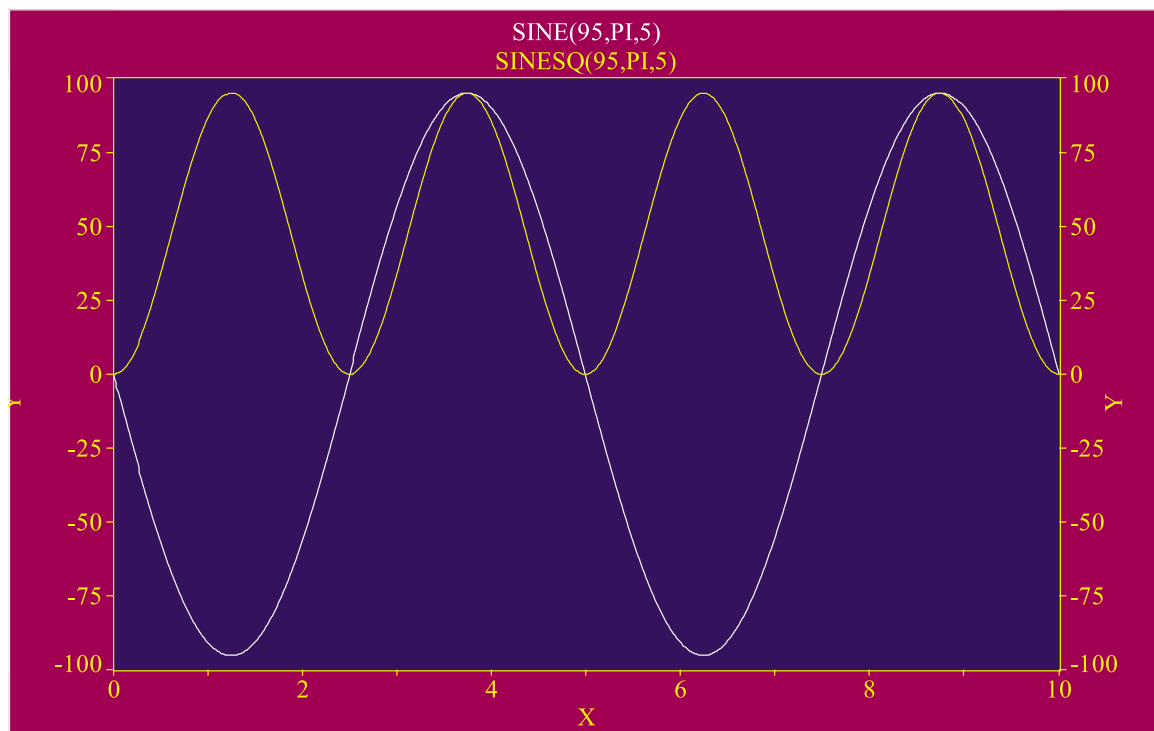
Equilibrium Conc $a \left(1 - \frac{c}{c+d} \right)$

Constraints $x > 0, b > 0, c > 0, d > 0, (c+d) \neq b$

Built-in Function **EQUILASBSC_(a,b,c,d)**

These equations are identical to the Equilibrium Peak Eqs. 8070 and 8071 in the peak group except that here the reaction begins at time $x=0$.

Waveform Equations



Amplitude: a

Phase: b

Wavelength: c

Sine $y = a \sin\left(\frac{2\pi x}{c} + b\right)$

Non-Intercept form is Eq. **8158**, intercept form is Eq. **8014**.

Sine² $y = a \sin^2\left(\frac{2\pi x}{c} + b\right)$

Non-Intercept form is Eq. **8159**, intercept form is Eq. **8015**.

Constraints $c > 0, 0 \leq b \leq 2\pi$

Built-In Functions **SINE(a,b,c)** and **SINESQ(a,b,c)** in PeakFit group.

General Equations

Power 8156 $y = ax^b$

8010 $y = a + bx^c$

Single Exponential 8157 $y = a \exp(-x/b)$

8002 $y = a + b \exp(-x/c)$

Two Exponential 8016 $y = a \exp(-x/b) + c \exp(-x/d)$

Straight Line 8160 $y = a + bx$ Minimize $\sum (y_i - \hat{y}_i)^2$

8161 $y = a + bx$ Minimize $\sum |y_i - \hat{y}_i|$

8162 $y = a + bx$ Minimize $\sum \ln(1 + |y_i - \hat{y}_i|^2)$

8163 $y = a + bx$ Minimize $\sum \ln\left(\sqrt{1 + |y_i - \hat{y}_i|^2}\right)$

TableCurve 2D Tutorials

- Fitting Approximating Functions
- Fitting Parametric Functions
- Non-Parametric Interpolation
- Noise Reduction
- Derivative Estimation
- Eigendecomposition Filtering
- Fourier Filtering
- Prediction and Forecasting

Fitting Approximating Functions

Interpolation and Root Finding

An approximating function is nothing more than an equation which is used to represent X-Y data. An approximating function can be used for interpolation, finding a Y-value for an X-value that is within the X range of the data or alternately, finding an X-root or roots for a Y-value that is within the Y data range.

Extrapolation

An approximating function can also be cautiously used for extrapolation, the process of finding a Y-value for an X-value outside the X range of the data or alternately, that of finding an X-root or roots for a Y-value that is outside the Y range of the data.

Derivatives and Integrals

Approximating functions can also be differentiated or integrated, both numerically and analytically, resulting in far more accurate areas under curves, inflection points, cumulative areas, and other aspects of data analysis than would be realized by working with the raw data. An approximating function is also the finest form of data smoothing in the time domain.

Linear Models

What is important in an approximating function is seldom the coefficient values derived from the fit, but rather the effectiveness of the fit itself. Equally is the number of coefficients of minor importance. For this reason, linear models are most frequently used for approximating functions. In the example that follows, the best candidate equations are linear models.

Creating The TableCurve 2D Data Table

Start TableCurve 2D. Select Start/Programs/TableCurve 2D v5.

The first step in using TableCurve 2D is to create an X-Y Data Table. TableCurve 2D can construct its data table from any of the following:

- ASCII files [X-Y, single column, multi-column]
- Excel v3-v5,95,97,2000 spreadsheet files [XLS]
- Lotus 123 spreadsheet files [WK4, WK3, WK1, WRK, WKS]
- Quattro Pro spreadsheet files [WB2, WB1, WQ1, WKQ]
- SigmaPlot worksheet Files [JNB, SPW, SP5, SPG]
- SPSS v8-v10 data files [SAV]
- Systat v8,v9 data files [SYS]
- dBASE III+ and dBASE IV database files [DBF]
- DIF files [X-Y, single column, multi-column]
- Binary files with repeating data blocks [with binary configuration]
- X-Y Data entered directly into the TableCurve or ASCII editors

TableCurve 2D offers a maximum table size of 65536 rows, and further offers an import-level averaging digital filter where up to 65 million values can be filtered into TableCurve 2D in a single step.



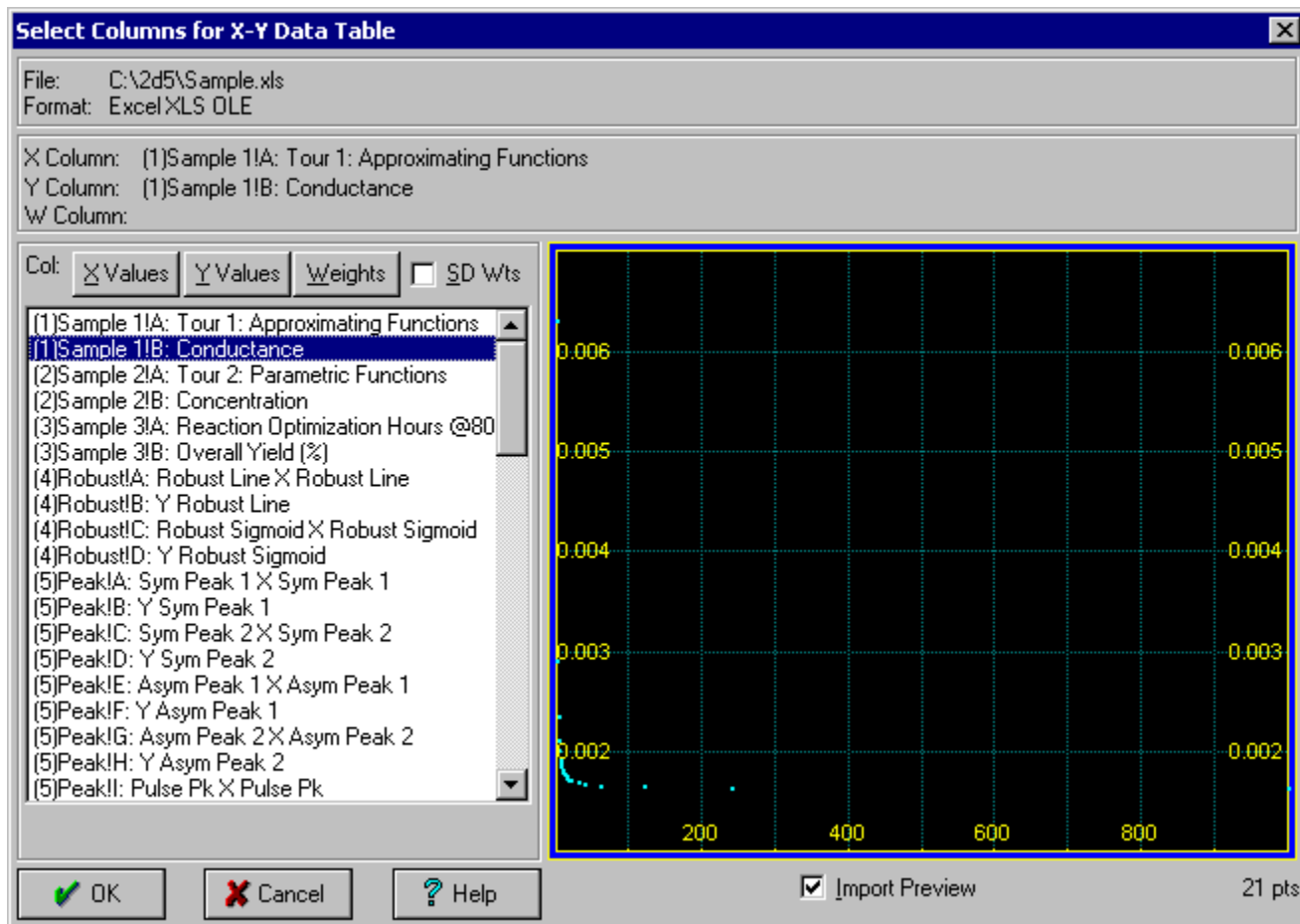
Select the File menu's Import option. Click on the Files of Type drop-down button and select Excel [xls] files. Select and open the file SAMPLE.XLS.

To aid in the identification of the columns, TableCurve 2D lists the string data existing in the first 100 rows

of the individual sheets or pages in the worksheet file. You must have at least one entry within the first 100 rows of a column in order for that column to be available for selection.

Select the first column with the label **Tour 1: Approximating Functions Temperature** to be used as the X-variable in the TableCurve 2D data table.

Select the column identified in the selection list as **Conductance** for the Y-variable. Check **Import Preview** to see a graph of the data that will be imported.



Note that the first two selections are automatically placed in the X and Y positions. You may double click on the column for the Y variable in order to immediately proceed with the import operation. You can also revise the initial X,Y selections as well as to specify a column to be used for the weights. The weights can be optionally imported as standard deviations.

Press **OK** to accept these choices. Press **OK** once again within the titles dialog to confirm the imported titles.

The selected columns of data are read and a summary is given for the read operation. 21 data points are read.



A background thread fitting may automatically occur as soon as the data is imported, depending on the **Background Thread Processing** option currently selected. For the purpose of this tour, we will ignore any background thread fitting that takes place.

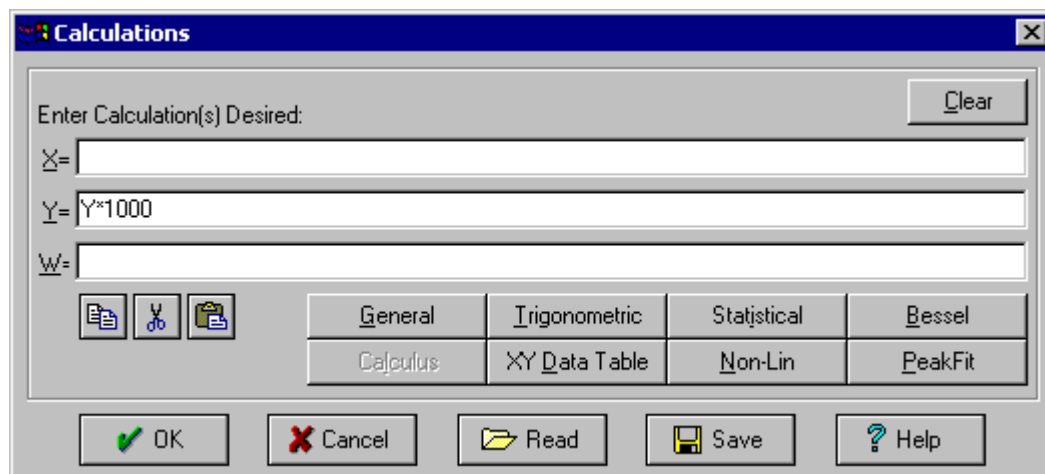
Once the data has been imported, TableCurve 2D proceeds to pre-scan the data table for basic statistics. At the conclusion of the import, a status window appears with these statistics and a graph of the data.

Calculation

The Y-values are in units of millimhos. We will use a calculation to adjust these units to micromhos.



Use the Data menu's Enter Calculation item and enter $Y*1000$ on the Y= line. Press OK and then answer Yes to immediately apply the calculation to the existing data table.



You will then see TableCurve 2D's standard Undo graph. This type of graph offers the means to graphically inspect the results of certain procedures before accepting the modifications.

Press OK to accept the Y-value changes introduced by the calculation.

Sectioning Data



Select the Data menu's Section Data item.



Click on the Logarithmic X Axis button in the graph's toolbar.

The data reveal a relatively constant conductance until absolute zero is approached. For the purpose of this tutorial, we shall assume that a temperature of 1°K was the experimental lower limit, but that an extrapolation is desired for values closer to absolute zero.

Individual points are toggled on and off simply by left clicking on them.



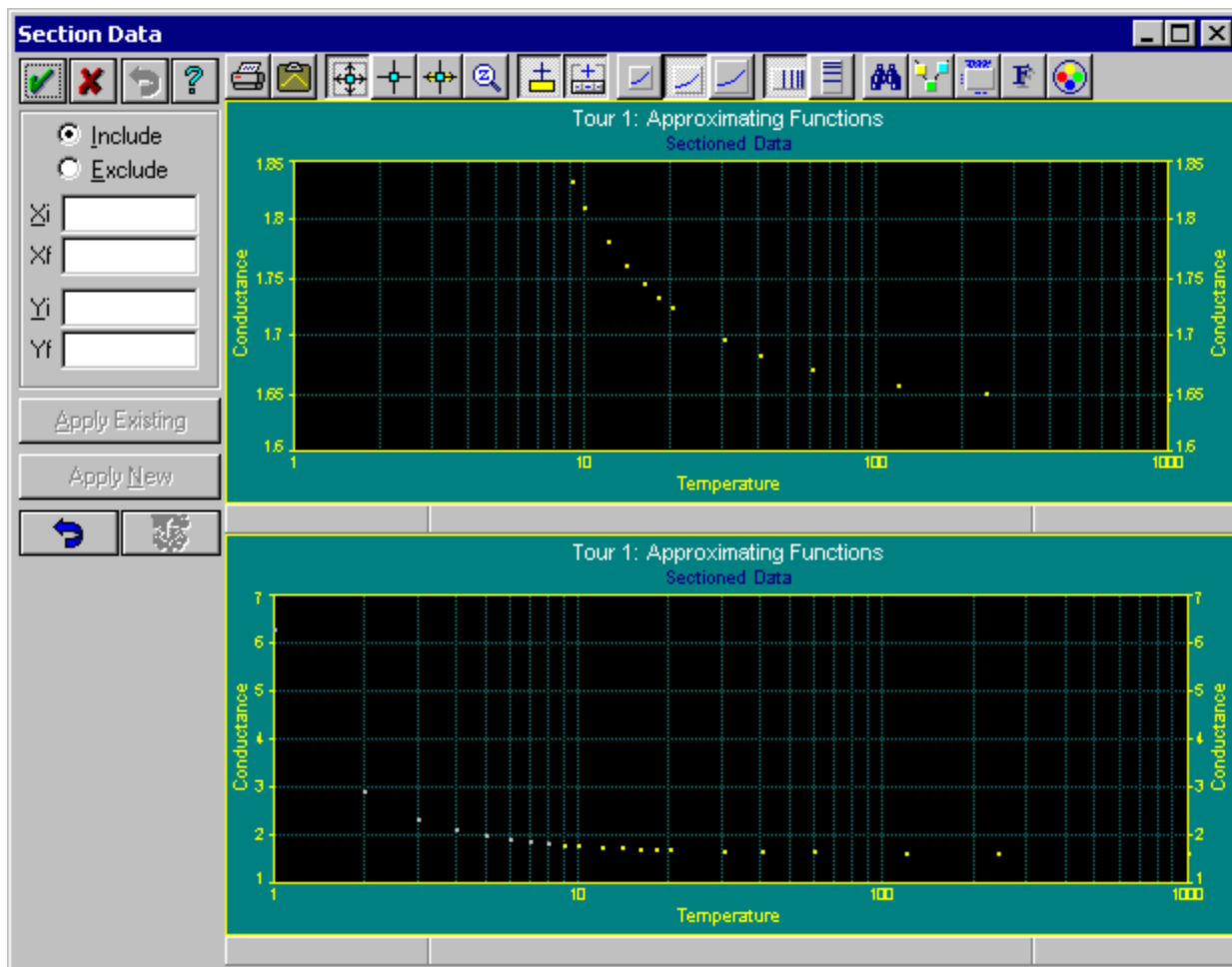
In X-Sectioning Mode, bands of data are rendered inactive by placing the cursor on the graph at the left border of the desired range and then pressing and holding the left mouse button down while sweeping to the right. Bands of data are reinstated for fitting by the reverse procedure of placing the cursor at the right border of the band and then sweeping the mouse to the left.



In XY Sectioning Mode, the mouse is used in the upper graph to enclose the x-y region of interest. To include the region that is to be graphically defined, click and hold down the left mouse button at one corner of the x-y rectangle in the upper graph and drag the rectangle so that it encloses the region desired. To exclude the region that is to be graphically defined, the same graphical procedure is followed

except that the right (rather than the left) mouse button is used.

The upper graph is scaled only for the active data and the lower graph is scaled for all data, regardless of the states of the individual points.



Experiment with turning bands of data on and off and with changing the status of individual points.



Press **Cancel** when done since we want to have all points active for the fitting. The status window should indicate 21 active points for the fitting.

Automated Processing

The TableCurve 2D engine processes the data table with up to 3491 built-in linear equations and up to 174 built-in non-linear equations in a highly automated single step.

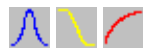
Because of the numerically intense nature of this processing, the time required for the fitting that follows will vary in relation to the computer's floating point processing speed. TableCurve 2D reports a floating point index (FPI) in the status window. The reference FPI is 100 for the original coprocessor-equipped IBM AT. A computer with an FPI of 50000 will process the data table 500x faster than this vintage AT.

A list of the full 3665 TableCurve 2D equation set is available on-line from the [List Equations](#) item in the

main TableCurve 2D Help menu.

Customizing TableCurve 2D's Fitting

TableCurve 2D offers a high level of optional customization.



The [Curve-Fit Peak Functions](#), [Curve-Fit Transition Functions](#), and [Curve-Fit Kinetics Equations](#) options have their own separate configurations which combine specific function selection with all non-linear controls appropriate to these types of functions.



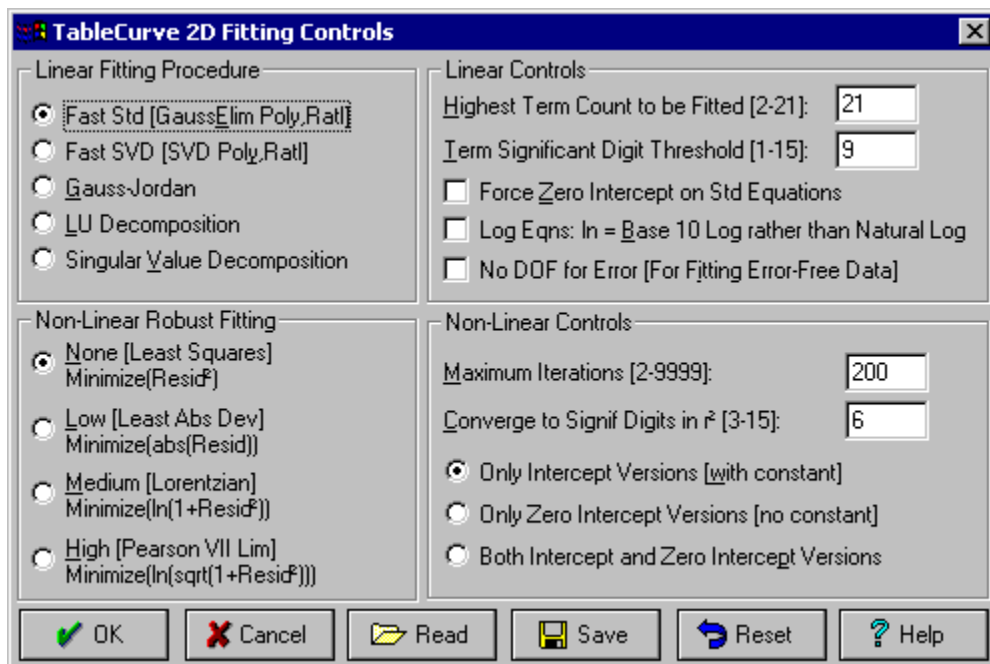
The [Curve-Fit Custom Equation Set](#) item fits the specific equation set constructed via the [Edit Custom Equation Set](#) option.



For all fit options except the peak, transition, and kinetic functions, the [Curve-Fit Preferences](#) option is used to set the various fitting controls.



Select the [Process menu's Curve Fit Preferences](#) option.



The default [Highest Term Count to be Fitted](#) of 21 accommodates all of TableCurve 2D's equations. The default [Term Significant Digit Threshold](#) of 9 means that an equation will be added to the list if each of its terms essentially makes at least a 1E-9 fractional contribution to the Y value. The [Force Zero Intercept on Std Equations](#) option automatically forces a zero intercept when fitting TableCurve 2D's 3205 standard (non-polynomial, non-rational) linear equations.

You can also force base 10 log functions and for handbook or generated error-free data, you can disable the degree of freedom normally allotted for the error of fit.

The [Fast Std](#) linear fitting procedure uses a mix of methods for resolving both the solution vector (coefficients) and the matrix inverse (errors and confidence limits). This offers the fastest possible fitting. The [Fast SVD](#) is similar except that singular value decomposition is used for the conventional fitting of higher order polynomials and rationals. This can sometimes result in more stable coefficients. You should

choose one of the other options only if you have extensive experience with matrix methods and wish to employ a single method for all numeric processing.

Press **OK** to close the preferences dialog.

Fitting the Linear Equations

In this tutorial we will be fitting a predefined equation set, that of all linear equations. At this point, we will initiate the automated curve-fitting.



Select the **Process** menu's **Curve Fit Linear Equations** option.

The entire curve-fit will occur automatically. You may cancel an automated fit in process by pressing the **Cancel** button. In the automated fitting, there is first a sums generation process. This is followed by the fitting of all 3491 linear equations.

When the fit is complete, you will see that over 2100 equations were added to the fitted set. A fast Pentium machine should require about a second for this fitting.

When the fit is concluded, enter the **Review** by clicking the **Graph Start** button.



As an alternative, you may press **OK** and select the **Graph Start** item in the **Review** menu. If you do not respond to the conclusion of the fit in 10 seconds, the **Review** is automatically started.

Reviewing the Fits

At the conclusion of the fit, equations with insignificant terms have been removed from the equation list. Other equations may be absent because they were not fitted. For example, there is no point in fitting an equation with a $\ln(x)$ term if there are negative x values in the data set.

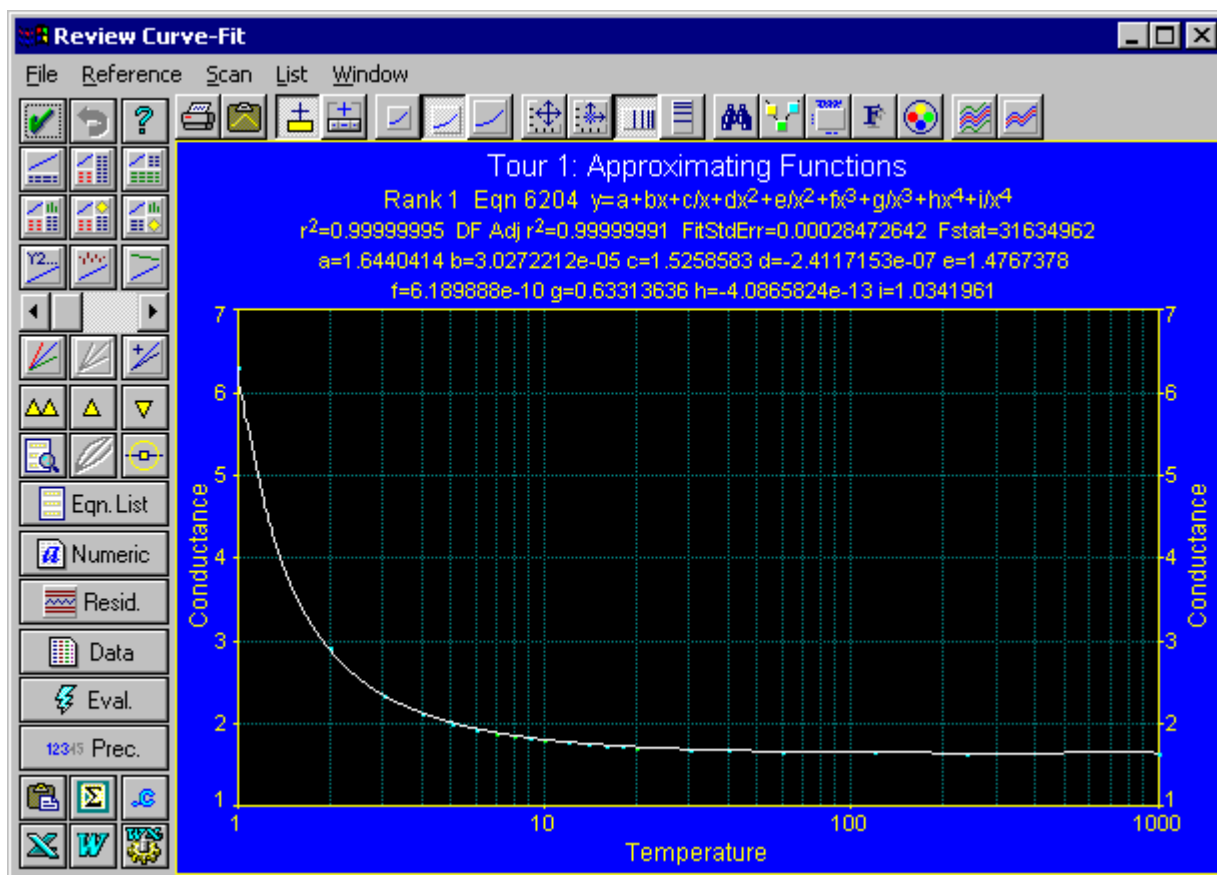


Click on the **Logarithmic X Axis** button in the **Curve-Fit** graph's toolbar.

Using the **List** menu of the **Curve-Fit** graph, or the **Sort** menu of the **Equation List**, select the **Sort by DOF Adjusted r^2** item.

This sorts the fitted equations by a degree-of-freedom adjusted coefficient of determination. For approximating functions, you will almost always want to sort by **DOF Adjusted r^2** or the **Fit Std Error**, both of which order the equations based upon a degree-of-freedom based least squares criteria.

The equation at the top of the list, #6204, is a balanced-order polynomial. The **Curve-Fit** Graph consists of a plot of the highest ranked equation:



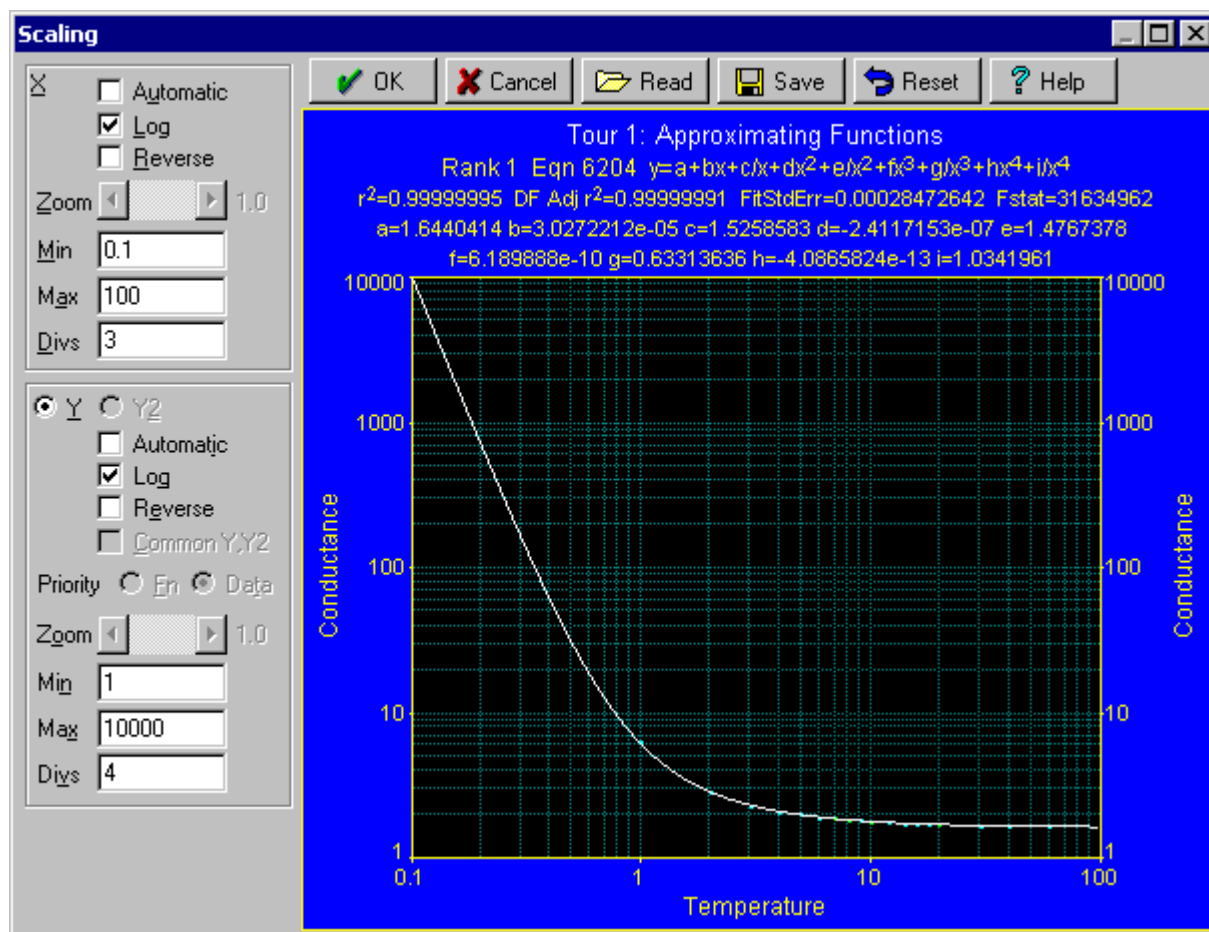
We will now choose from among the candidate equations to select the one best suited to this application. Note that the [Equation List](#) contains fewer equations than the number successfully fitted. The absent equations are rationals with an undefined region or pole in the X-range of the data that have been filtered out of the list by the default [Rationals Defined Xmin-Xmax](#) option. You will not normally be interested in such rational equations. All other equations will initially be present.

We will first set the scaling for the extrapolation.



Click on the **Modify Graph Scaling** button in the graph's toolbar. For the X-axis, be sure the **Log box** is checked, **uncheck the Automatic box**, and enter **0.1** for the minimum, **100** for the maximum, and **3** for the divisions. For the Y-axis, check the **Log box** and **uncheck the Automatic box**, and enter **1** for minimum, **10000** for the maximum, and **4** for the divisions.

The graph should appear as follows:



Click **OK** to accept the Scaling settings.

Equation #6204 Balanced Order Polynomial

The highest ranked equation is a balanced order polynomial. Such polynomials contains both standard and inverse terms. The next two equations in the list are also balanced order polynomials. The next five ranked equations consist of non-linearly fit constrained rational models.

Explore the first fifteen ranked fits by selecting the equations within the **Equation List** window.

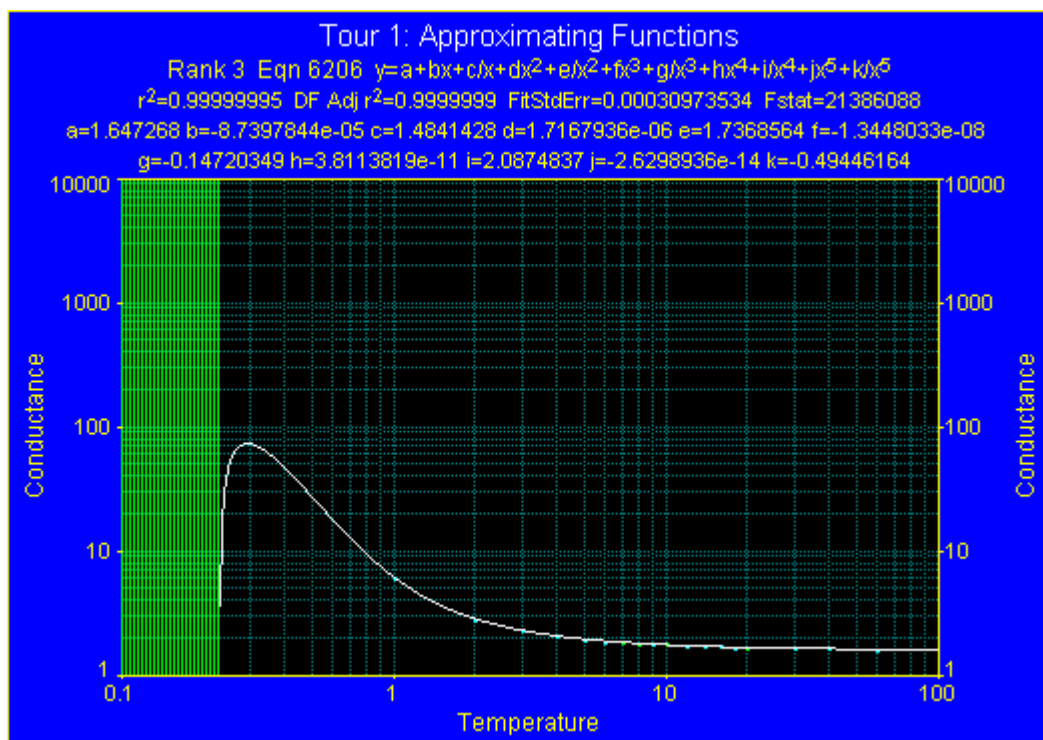


You can also use the arrow buttons in the Curve-Fit Graph's control panel to step through equations.

If the Equation List window is active, you can also use the cursor keys.

If the Curve-Fit graph window is active and you have a scroll-wheel mouse, you can scroll the equations using the mouse wheel.

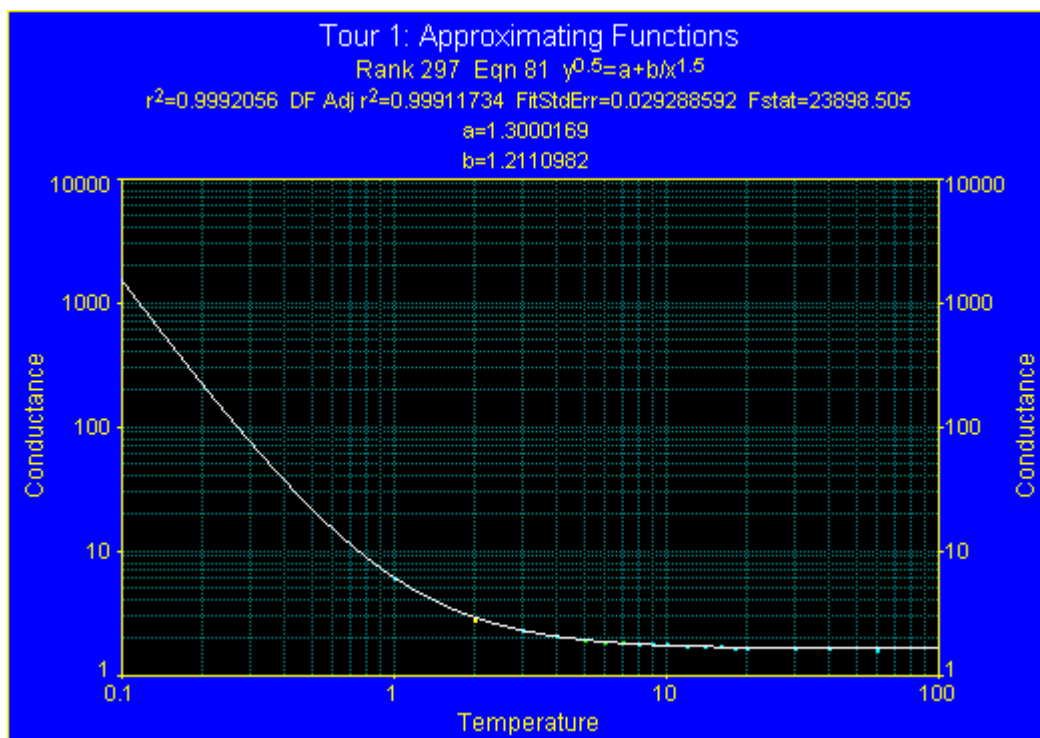
Note the vertical lines that are drawn with the third ranked equation. These lines represent an undefined condition or math error. Here, the function is merely undefined on a log scale since it evaluates to negative values.



Simple Equations Filter

All of the initial equations in the list are linear equations. To see how well TableCurve 2D's simple two-parameter linear equations fared with this data set, we will apply a simple filter.

Using either the List menu of the Curve-Fit Graph or the Filter menu in the Equation List window, select the Simple Equations option.

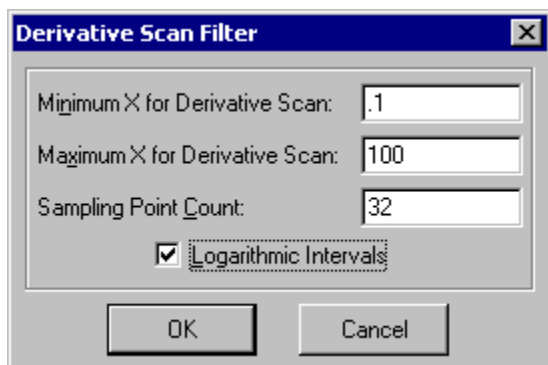


Although there are nearly 300 equations with a better DOF adjusted r^2 , the best of the simple 2 parameter models offers a very respectable fit.

Derivative Filtering

The filter we want is one that will assure a continuously increasing trend in the region of extrapolation.

Using either the List menu of the Curve-Fit Graph or the Filter menu in the Equation List window, select the Set X-Range for Derivative Filtering option and enter 0.1 for the minimum X value and 100 for the maximum X value. Also check the Logarithmic Intervals check box and then click OK to exit the dialog.



This sets the scan conditions for the Continuous Trend First Derivative and Constant Sign in First Derivative filters but does not automatically select one of them.

Using either the List menu of the Curve-Fit Graph or the Filter menu in the Equation List window, select the Continuous Trend First Derivative option.

This filtering may take a few seconds since there are 2100 equations and each equation's first derivative will be evaluated at 32 logarithmically-spaced X-values to determine if a continuous trend, either increasing or decreasing, exists across this range.

When this filtering is complete, you should see that the list has been reduced to 696 equations. The two highest ranked equations remain at the top of the list. The rank always reflects all fitted equations as ordered by the current goodness-of-fit sort criteria. Note that only five of the top twenty ranked equations offer the desired continuous trend in the specified region.

Step again through the highest ranked equations and note that all have the desired trend for the extrapolation. When finished, return to the top ranked equation.

The FP Item for Scientific Programmers: Execution Speed

The third item in the equation list is an estimate of the floating point complexity of the equations, based on TableCurve 2D's own code generation option. For approximating functions, the "simplest" equation may well be the fastest executing one, not the one with fewest coefficients. TableCurve 2D will automatically generate code with 15 or 18 digits of precision in the coefficients. An FP value of 1 corresponds to an Intel FPU addition. An equation with an FP of 45 should require about three times as long to execute as one with an FP of 15. Note the significant variation in execution speed relative to the number of parameters.

SE Colored Residuals

Note the coloring of the points based upon the number of standard errors represented by the residuals. If you are using a predefined color scheme, the points less than 1 standard error from the curve will be blue in color. Green points are between 1 and 2 standard errors, while yellow points are between 2 and 3 standard errors, and red points are beyond 3 standard errors.

Confidence and Prediction Intervals

The confidence intervals are a measure of how accurately the average curve for repeated experiments is determined. The prediction intervals measure how accurately an individual curve is determined. The prediction bands will always be wider than the confidence bands. For approximating functions, the prediction intervals, which deal with the confidence about the predicted Y values for the next experiment, will be the main ones of interest.



Click on the Set Confidence/Prediction Intervals, %Confidence button in the graph's toolbar. Be sure the Prediction type of interval is checked, the Confidence type of interval is unchecked, and that the 99% level is checked. Click OK.

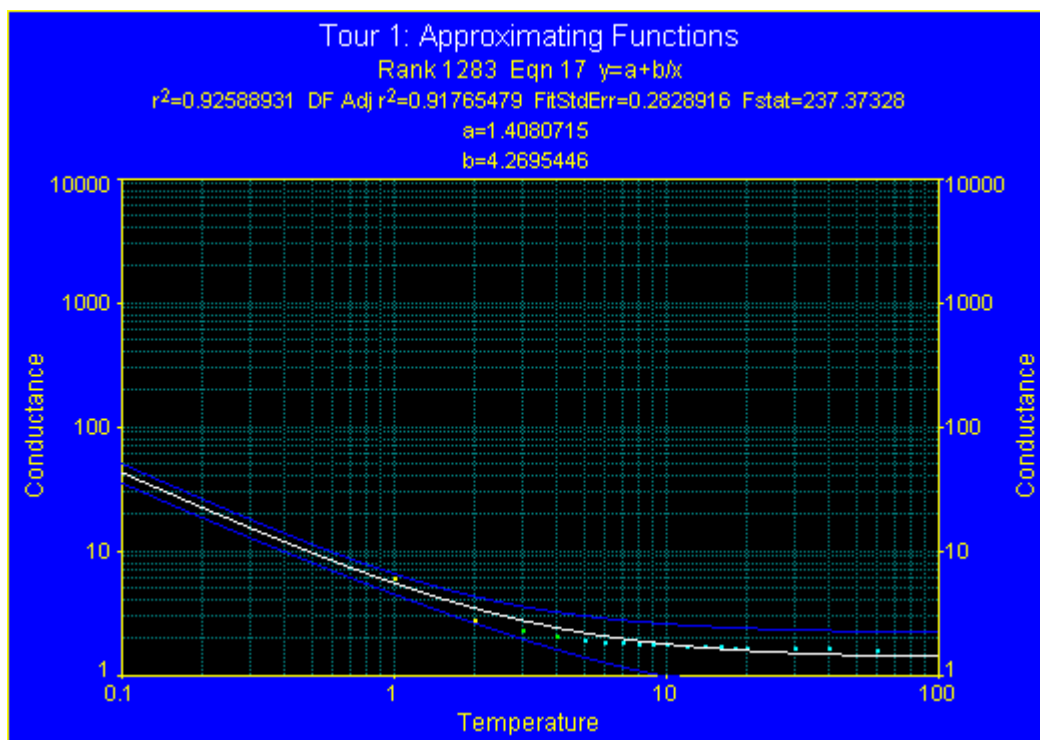
The 99% prediction interval is a rather demanding confidence interval. Still, the intervals in the region of extrapolation are very tight and are very close to the actual curve.

Searching A Specific Equation

To observe wider prediction intervals, we will inspect an equation whose fit produces significantly larger errors.



Click on the Search for Specific Equation button, enter equation 17, press OK, and observe the wider intervals about the fitted curve.



Click on the **Show Confidence/Prediction Intervals** button to toggle off the intervals.



Click on the **double up arrow** button to move back to the top of the list.

Note that the computation of intervals is highly sensitive to machine precision, especially outside the X data range. When confidence intervals errors occur in the data range of interest, you should assume the equation is only marginally stable in this region.

Also note that careful judgment is required in extrapolations. The tight prediction bands of a given equation hardly match those of different but also highly-ranked equations.

Fit References

The question at this point is no longer whether Equation 6204 is a suitable approximating function, but how far such a model can be trusted in extrapolations toward absolute zero. In this data set, as in any form of extrapolation, it is important to remember that the equation is based only upon the presence and magnitude of physical phenomena in the range of the data. For example, wild forces could come into play at temperatures below 1°K that have little presence above, where the actual experimental data ends.

To determine how far to trust this extrapolation, we will now set this highest ranked fit as a reference.



Click the **Add Current Equation as Reference** button (or the **Add Current Equation as Reference** in the Reference menu).

Initially, the reference and current curve will be identical, overlapping one another.

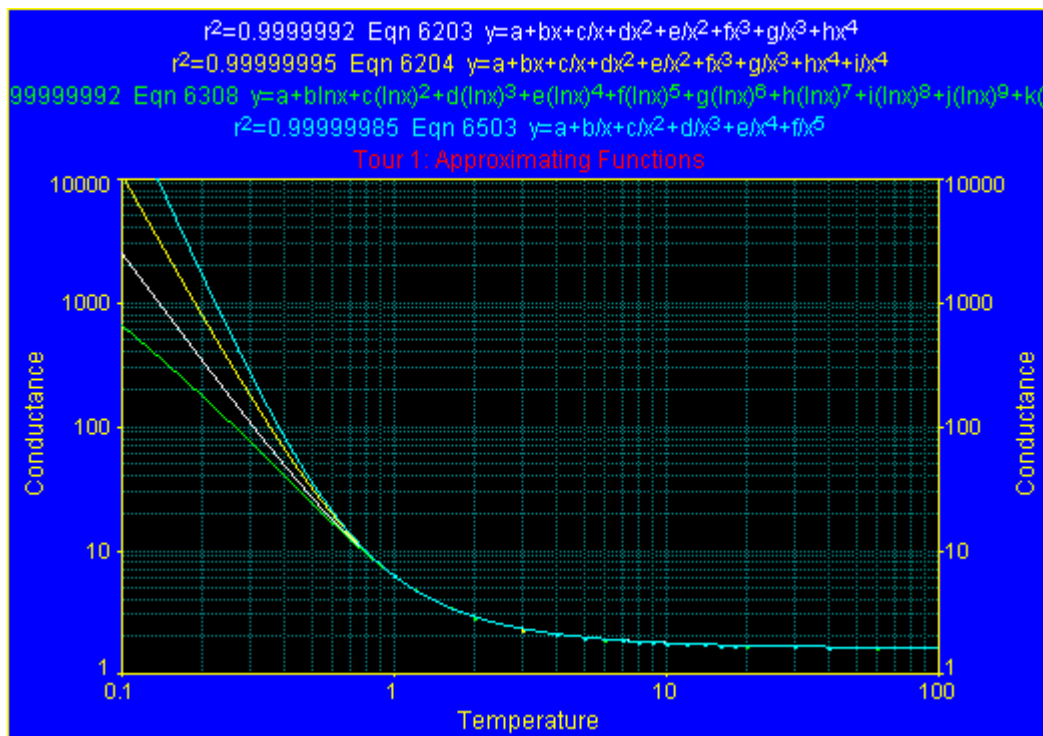


Select the third equation in the list, **Equation 6308** and again click the **Add Current Equation as Reference** button.



Select the fifth equation in the list, Equation 6503 and again click the **Add Current Equation as Reference** button.

Select the sixth equation in the list, Equation 6203.



Note that there is a considerable divergence in the extrapolations among these four excellent fits.



Use the arrow buttons or the equation list to inspect some of the other equations against this set of references.



When done, use the double up arrow button (or select Equation 6204) to return to the highest ranked equation in the list.



Use the **Toggle Display of Equation References** button to disable the display of the references.

Evaluation

The **Eval** item is used to open a full-featured numeric evaluation of the equation currently selected. An evaluation table will contain a history of all numeric operations requested. We will use both a single entry and an automatic generation procedure for adding information to this table.



Eval.

Use the **Eval** button to open TableCurve 2D's Evaluation procedure.



Enter .5 in the X field and click on the **y=f(x)** button for a simple function evaluation.



Enter .1 in the X field and again click on the **y=f(x)** button.

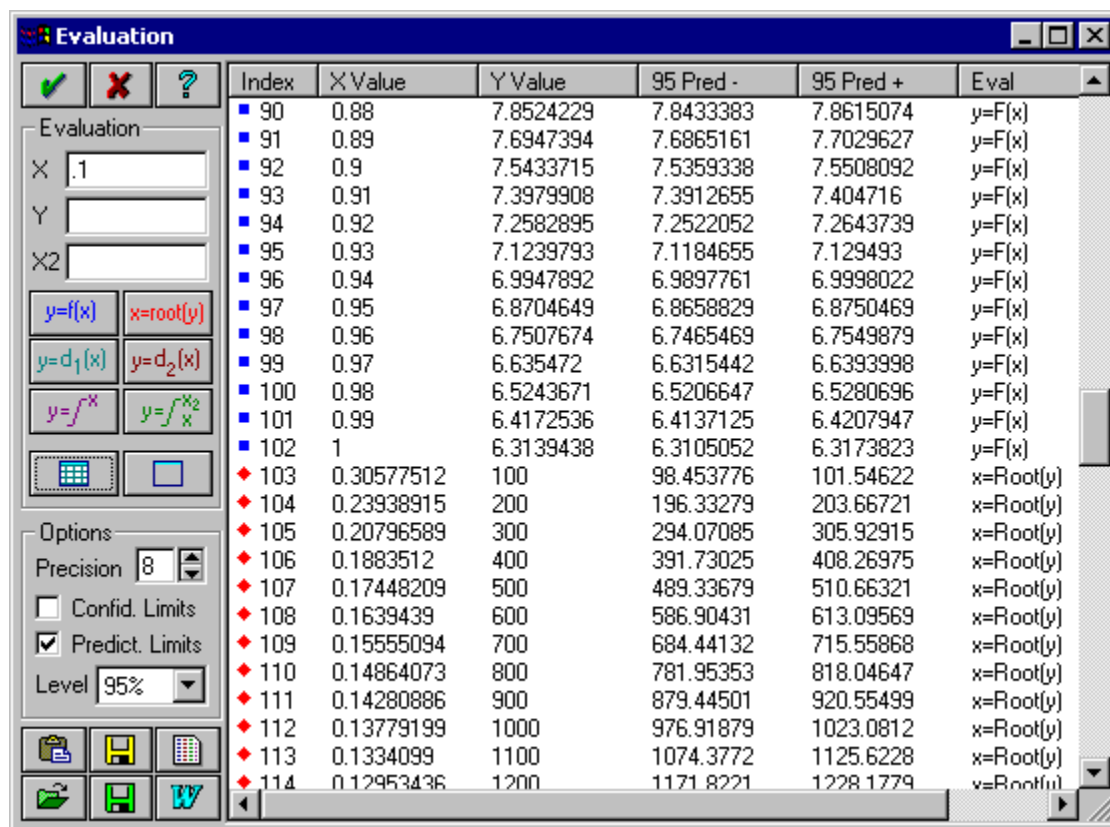
 Successively enter **100**, **1000**, and **10000** in the Y field followed each time by pressing the **x=root(y)** button.

 Click on the **Generate Table** button. Be sure the options are set to **Generate the data To Evaluation Table** where **X is Input, Y=Fn at X** before pressing **OK**. Enter **.01** as the initial X value, **.01** as the increment, **1.0** as the final X-value, and then press **OK**.

You will see 100 function evaluations added to the table.

 Again press **Generate Table** but now select the **Y is Input, X=Root of Fn at Y** option. Enter **100** as the initial Y-value, **100** as the increment, and **10000** as the final Y-value and press **OK**.

You will see 100 roots added to the table.



The screenshot shows the 'Evaluation' window with a table containing 114 rows. The first 100 rows are function evaluations (y=F(x)) and the next 14 rows are roots (x=Root(y)). The table has columns for Index, X Value, Y Value, 95 Pred -, 95 Pred +, and Eval. The 'Eval' column contains the equation being evaluated for each row.

Index	X Value	Y Value	95 Pred -	95 Pred +	Eval
90	0.88	7.8524229	7.8433383	7.8615074	y=F(x)
91	0.89	7.6947394	7.6865161	7.7029627	y=F(x)
92	0.9	7.5433715	7.5359338	7.5508092	y=F(x)
93	0.91	7.3979908	7.3912655	7.404716	y=F(x)
94	0.92	7.2582895	7.2522052	7.2643739	y=F(x)
95	0.93	7.1239793	7.1184655	7.129493	y=F(x)
96	0.94	6.9947892	6.9897761	6.9998022	y=F(x)
97	0.95	6.8704649	6.8658829	6.8750469	y=F(x)
98	0.96	6.7507674	6.7465469	6.7549879	y=F(x)
99	0.97	6.635472	6.6315442	6.6393998	y=F(x)
100	0.98	6.5243671	6.5206647	6.5280696	y=F(x)
101	0.99	6.4172536	6.4137125	6.4207947	y=F(x)
102	1	6.3139438	6.3105052	6.3173823	y=F(x)
103	0.30577512	100	98.453776	101.54622	x=Root(y)
104	0.23938915	200	196.33279	203.66721	x=Root(y)
105	0.20796589	300	294.07085	305.92915	x=Root(y)
106	0.1883512	400	391.73025	408.26975	x=Root(y)
107	0.17448209	500	489.33679	510.66321	x=Root(y)
108	0.1639439	600	586.90431	613.09569	x=Root(y)
109	0.15555094	700	684.44132	715.55868	x=Root(y)
110	0.14864073	800	781.95353	818.04647	x=Root(y)
111	0.14280886	900	879.44501	920.55499	x=Root(y)
112	0.13779199	1000	976.91879	1023.0812	x=Root(y)
113	0.1334099	1100	1074.3772	1125.6228	x=Root(y)
114	0.12953436	1200	1171.8221	1228.1779	x=Root(y)

This generation option can also be based upon data that is read from a column in an ASCII, Lotus, Excel, Quattro, SigmaPlot, SPSS, Systat, dBase, or DIF file.

When **Confid. Limits** or **Predict. Limits** is checked, two additional columns containing the confidence or prediction intervals are added to the evaluation table for both function evaluations and roots.

The entries in the evaluation table can be printed, copied to the clipboard, or saved to file in an ASCII, Excel, Lotus, SPSS, or Systat format. The data can also be listed, streamed to MS Word, and saved as an RTF report. The evaluation sequence can also be saved to disk and recalled at some future time for use with a different set.

  Select the second ranked equation. Note that the Evaluation table is automatically updated.

Return to the first ranked equation. Click **OK** in the Evaluation window to close the Evaluation procedure.

Output Options

Each of the key Review windows has its own output. If it is a text or equation list window, you can save the window's contents to an ASCII, WK1, or RTF file or stream the contents to MS Word.



The Curve-Fit Graph window offers a printed graph, including a half-page graph with half-page numeric summary

The Curve-Fit graph's File menu offers the following output:

- Three formats of an Excel XLS worksheet file ranging from all possible curve-fit information, including an Excel representation of the equation, to a simple two column worksheet with generated XY information
- A Lotus WK1 spreadsheet file with over 12 named graphs, full curve-fit statistics, observed and generated data sections, and a WK1 representation of the curve-fit equation
- A SigmaPlot graph file with up to four publication quality graph pages and an optional SigmaPlot transform generation.
- A Harvard Graphics chart of the curve fit graph
- Program Code Generation in C, Pascal, Fortran 77 or 90, and Visual Basic or QBasic languages
- Generated ASCII output with full confidence information and extensive format control
- An ASCII file containing the parameters to full precision with optional identifiers and descriptive information
- An ASCII file containing the covariance matrix from the fit
- A TableCurve 2D User Defined Function based upon the fitted equation

Custom Titles

The main Edit menu's **Modify Titles** option is used only to set basic titles for the data table. These titles are used in all non-graphical export options.



The **Modify Graph Titles** button in the toolbar of each of the program's graph is used to create customized titles for the graphs.

Click the **Modify Graph Titles** button in the Curve-Fit Graph's toolbar. Enter **(K)** after Temperature and **(mho)** after Conductance.



Place the cursor before the K in the X Title and click the **Symbol** button. Select the ° character from the Symbol set.



Place the cursor before the mho in the Y title and again press **Symbol**. Select the μ in the Symbol set.



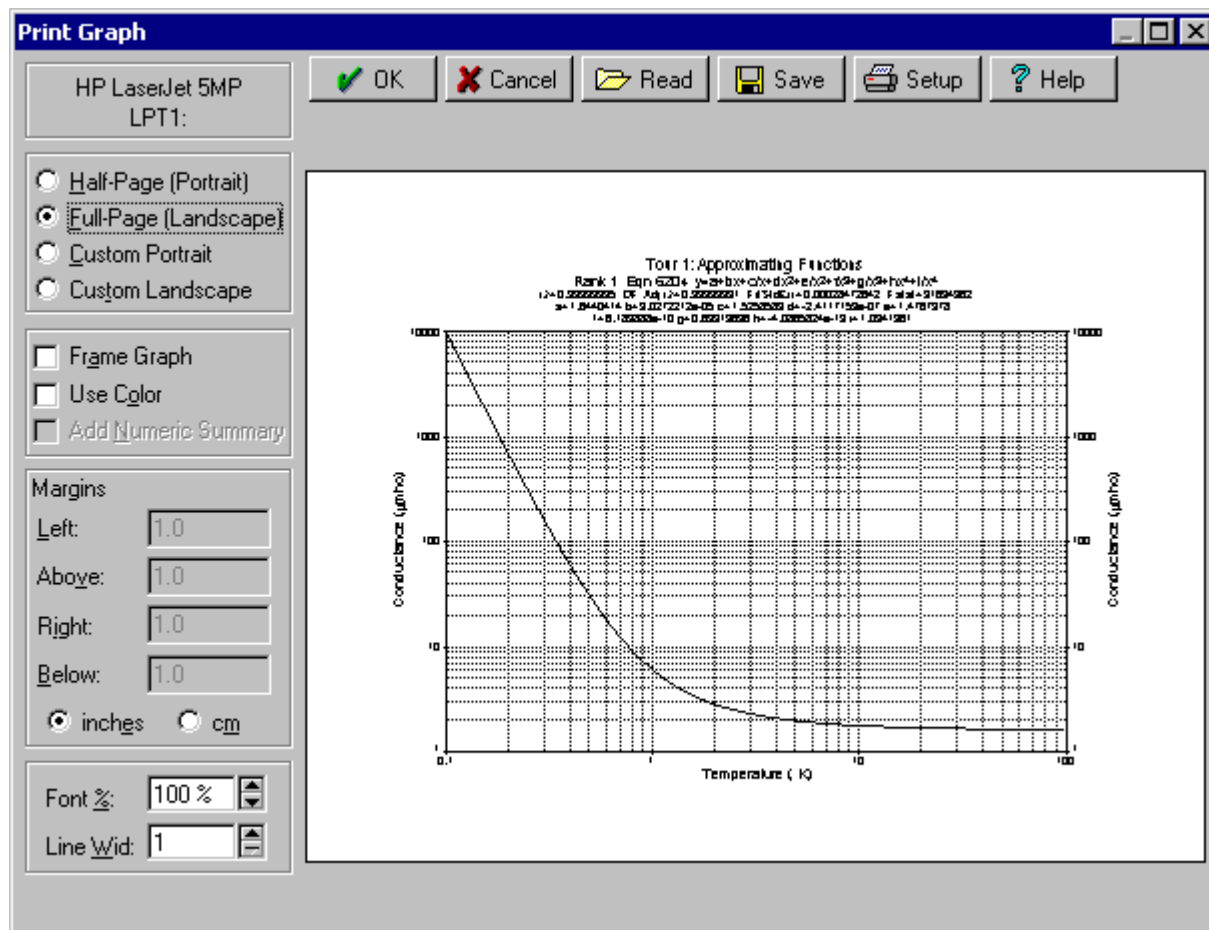
Use the **Copy** and **Paste** buttons to create an identical Y2 title. Press **OK** to close.

Printer Configuration

For the last part of this tutorial, you may need to configure your printer. This can be done using the **Setup** button in the Print dialogs.

Graph Print

 Select the **Print Graph** button in the Curve-Fit Graph's toolbar. Print preview is automatic. Choose the format and options desired and press **OK** to confirm the print.



Note that you can adjust the font scaling and the line width in printed output.

Code Generation

If you do any scientific programming, you should find the code generation capabilities of TableCurve 2D to be a strong asset.

 Select the **File Menu's Code Generation** option and then further select the **Full Test Code** option and the program language of choice. You must also specify the function, subroutine, or procedure name for the function code containing the fitted equation. Press **OK** and then input **temp** as the file name.

The correct language extension will be appended. The code is then generated and an ASCII listing of the file follows as a confirmation.

Inspect the generated code and then close the list window.

All 3665 of the built-in equations are eligible for code generation. The code is concise, even for polynomials and rationals, and double precision (15 significant digits) is generally used. If the equation contains the error function, its code is also automatically generated.

Spreadsheet Export



The Excel export option offers three different levels of spreadsheet content, three different Excel formats, and a full generated data section with curve-fit equation.

The WK1 (Lotus-123) export option is similar to the full content Excel export except that the worksheet can optionally contain named graphs.



If you are an Excel, Lotus, or Quattro Pro user, select the **Save Lotus 123 or **Save Excel** option in the Curve-Fit graph's File menu. Press **OK** to accept the defaults and then enter the name **temp** and press **OK**.**

If possible, the worksheet will contain a Lotus or Excel representation of the equation stored as a string. You will need to remove the leading string character to activate the formula, and you will need to copy it to any additional cells desired in the column. The X values are entered in the previous column. If you are using Excel v5 or higher, you can also generate Visual Basic code which can be inserted as a module into the worksheet. This makes the TableCurve 2D generated function accessible to cells just as if it were an Excel built-in function.

Note that you have full control over the range and size of the generated data section of this spreadsheet. For the Excel export, you may elect to create a worksheet with only the generated information, including confidence and prediction limits, or with just the X and Y values.

SigmaPlot Output



If you are a SigmaPlot user, select the **Save SigmaPlot option in the Curve-Fit graph's File menu. Select the SigmaPlot format of choice and press **OK**. Enter the name **temp** and press **OK** to generate the SigmaPlot file.**

This option generates a full-featured SigmaPlot worksheet containing publication quality graphs.

With the SigmaPlot output, you also have control over the range and size of the generated section of the worksheet. For most equations, you may also have TableCurve 2D generate a SigmaPlot XFM transform file that uses the same file prefix but has an XFM extension.

Exiting TableCurve 2D



Click **OK in the Curve-Fit Graph to exit the Review.**

To exit TableCurve 2D, either close the main TableCurve 2D window, or use the **Exit item in the main File menu.**

Fitting Parametric Functions

This tutorial focuses upon the task of finding a parametric model for a data set. In a parametric model, the main emphasis is upon solving for the parameters that are intimately associated with features within the data. Unlike approximating functions intended only for interpolations and extrapolations, a parametric model is very sensitive to parameter count and to interactions between the parameters.

Parametric Functions

A parametric model need not have an underlying theoretical foundation, although such is frequently the case. When there is a theoretical underlying model, each of the parameters represents a physically or theoretically real quantity such as energy, concentration, time, temperature, and so forth.

A parametric model with an underlying theoretical foundation relates the dependent variable Y to the independent variable X via a postulated physical relationship between them.

Often a parametric model is used not for describing an underlying theoretical model but rather for characterizing the unique features within a data set. The data set used in this second tour is of this type. Here features of the data are known to have a direct relationship with the underlying physics, chemistry, biochemistry, physiology, or human factors, but there is no quantitative underlying theoretical model.

This type of analysis, which treats the system producing the data as a "black box", results in parameters that can be used to infer trends or characteristics. No better example exists than human biochemistry whose complexity often precludes the availability of a theoretical model. The example that follows should clearly demonstrate the nature of this type of parametric curve-fitting.

Importing A Pharmacological Data Set

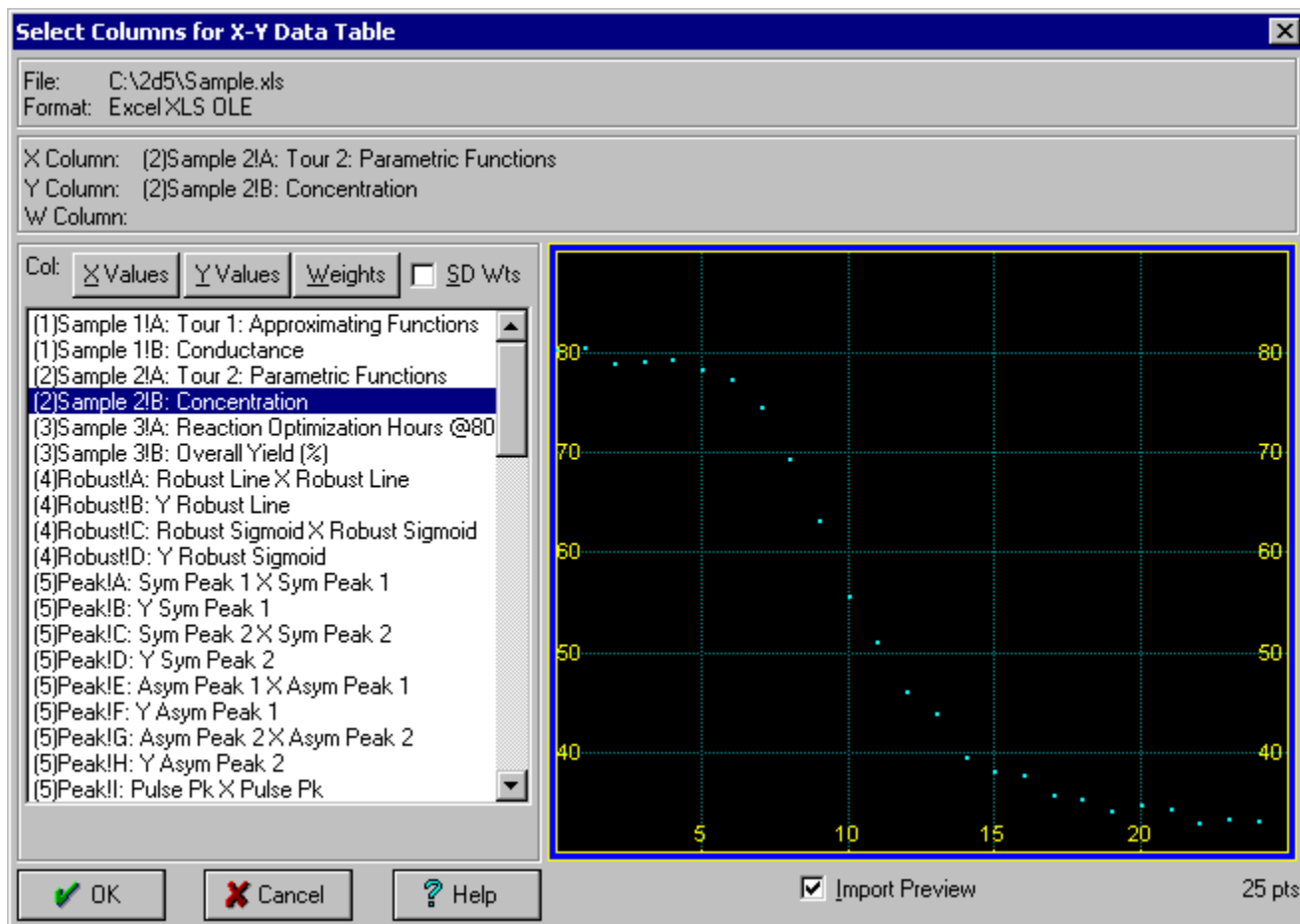
Start TableCurve 2D. Select **Start/Programs/TableCurve 2D v5**.



Select the File menu's **Import** option (or use the **Import** button in the main toolbar). If **Excel [xls]** files are not shown, click on the **Files of Type** drop-down button and select **Excel [xls] files**. Select and open the file **SAMPLE.XLS**.

To aid in the identification of the columns, TableCurve 2D lists the string data existing in the first 100 rows of the individual sheets or pages in the worksheet file. You must have at least one entry within the first 100 rows of a column in order for that column to be available for selection.

Select the third column with the label **(2)Sample2!A: Tour 2: Parametric Functions** to be used as the X-variable in the data table. Select the fourth column identified in the selection list as **(2)Sample2!B: Concentration** for the Y-variable. Check **Import Preview** to see a graph of the data that will be imported.



Note that the first two selections are automatically placed in the X and Y positions. You may double click on the column for the Y variable in order to immediately proceed with the read operation. You can also revise the initial X,Y selections as well as to specify a column to be used for the weights. The weights can be optionally imported as standard deviations.

Press OK to accept these choices. Press OK once again within the titles dialog to confirm the imported titles.

The selected columns of data are read and a summary is given for the read operation. 25 data points are read.



Note that a background thread fitting may automatically occur as soon as the data is imported, depending on the [Background Thread Processing](#) option currently selected. For the purpose of this tutorial, we will ignore any background thread fitting that takes place.

X=0,Y=0 Data

The first data pair, which has an x value of 0.0, is included in the data set. Although TableCurve 2D substitutes 1E-25 for 0.0 in an effort to keep all logarithmic and inverse equations available, it is usually best to exclude all x=0 points (and y=0 as well if fitting $\ln y$ or $1/y$ equations) unless the point has real significance. As you will note here by looking at the first few data points, this x=0 measurement is not needed because the upper plateau is represented adequately by the subsequent points. As such, we will first exclude this initial point prior to fitting.

TableCurve Editor

TableCurve 2D offers two numeric editors and also the means to graphically edit data. The [TableCurve Editor](#) offers a simple spreadsheet-like form of editing, while the [ASCII Editor](#) offers the ability to edit the data table in a Notepad-style of editor.



In the Data menu's [Section Data](#) option, points can be toggled on and off simply by clicking on them. For this tutorial, we will exclude the point in the TableCurve Editor.



Select the Edit menu's [TableCurve Editor](#) option. Check the **Ex (Exclude)** box for point #1 and press **OK**. You may answer **No** to the prompt for saving the revised data.

The TableCurve 2D status window will now show 24 rather than 25 active points and in the graph, the first point will be shown in the inactive color.

At time=0, the species under study had a concentration of about 80 units. This corresponds to the level in the bloodstream after injection. After some hours, the level begins to drop and goes through an S-shaped transition to a final value near 35 units that represents the level of species normally present.

Transition Functions

Such transitions, while frequent in pharmacology, are found in virtually every science. Often they are positive in their slope rather than negative. The equations describing this type of data trend are often referred to as transition, S-shaped, or cumulative functions.

In this example, the models being fitted will not address the underlying biochemistry. What is important is that the models each offer the following parameters:

- Concentration at the upper or lower plateau
- Height of the transition in concentration units
- Time value for the center of the transition
- A transition width, slope, or steepness factor that characterizes how rapidly the transition occurs
- If needed, a parameter that characterizes the asymmetry present in the transition

Note that such parametric models offer a meaningful way to map the essential features of the data. The scientist can then use this type of model in repeated experiments to see how various biochemical factors might affect one or more of these fitted parameters.

Fitting The Transition Functions

You can use the Help menu's [List Equations](#) item to display the full equation set. The transition functions are near the bottom of the list within the non-linear equations. Specific mathematical properties of these models are given in the [Transition Functions](#) section of the help system and in the [NONLINEQ.PDF](#) Adobe Acrobat document included in the product distribution.

TableCurve 2D offers 29 built-in non-linear transition function equations, zero-intercept and intercept forms of 14 different transition models and an impulse model. Of the 15 individual models, 5 produce a symmetric transition and 10 produce an asymmetric one.

For this example, we wish to fit only these non-linear transition functions. Linear models, particularly rational equations, can fit this data very effectively, but they are approximating functions only, and the coefficients will not yield the desired parameters which describe the essential features within the data.

In the intercept form of the transition functions of interest, the *a* term represents the lower plateau, *b* the height of the transition, *c* the center of the transition, and *d* describes the steepness of the transition.

Customizing TableCurve 2D's Transition Function Fitting

TableCurve 2D offers separate customizations for fitting its peak, transition, and kinetic functions. Each of these configurations places all pertinent controls and function selection information in a single place. These options also offer an [AI Expert](#) option which will scan the profile of the data and suggest recommended equations.

 **Select the Process menu's Curve-Fit Transition Functions and then click the AI Expert button.**

The [AI Expert](#) option suggests that it would be a waste to bother with symmetric transitions, since the data were found to have a clear asymmetric nature. This option also suggests the fitting of only the intercept version of the functions since neither plateau is anywhere near zero.

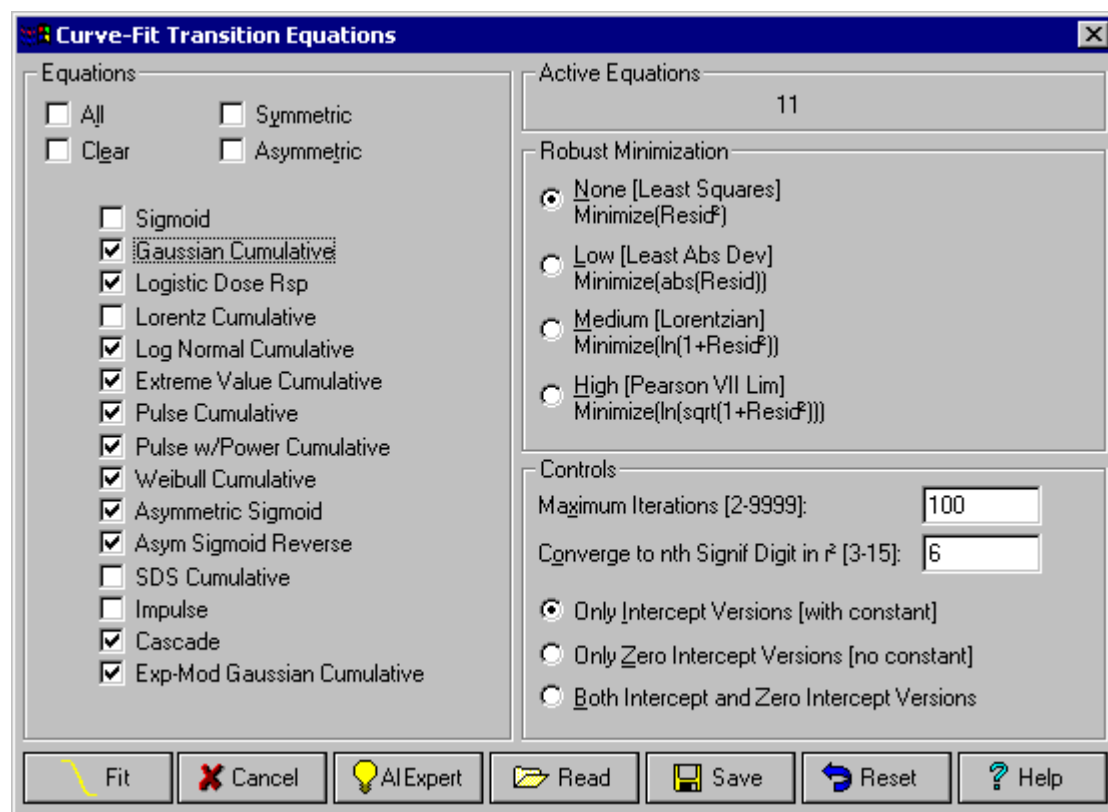
Non-linear fitting is iterative, and as such, the maximum number of iterations permitted for a non-linear equation, as well as the convergence criteria, are selectable. The default value of 100 should be sufficient for all except complex UDFs with poor starting estimates.

The default convergence precision of 6 means that the r^2 must be unchanging in the sixth significant figure for 5 consecutive iterations to signal convergence.

Since there are no apparent outliers in this data set, no reason to believe that errors are other than normally distributed, and only a modest Y-range, there is no reason to use a robust minimization.

For the purpose of this tutorial, we will add a symmetric transition function.

Click on the Gaussian Cumulative function.





Press **Fit** to initiate the fitting of these functions.

This should require less than a second on a Pentium machine. Ten equations are successfully fitted.

When the fit is concluded, enter the Review by clicking the **Graph Start** button.



As an alternative, you may press **OK** and select the **Graph Start** item in the Review menu. If you do not respond to the conclusion of the fit in 10 seconds, the Review is automatically started.

Reviewing The Transition Equations

Using the **List** menu of the **Curve-Fit** graph, or the **Sort** menu of the **Equation List**, select the **Sort by F-statistic** item.

For parametric functions, you will almost always want to use the F-statistic sort method which orders the equations based upon how accurately a given model can be said to describe the data.

We will first insure that we are using the simpler notational format for non-linear models. Some of the non-linear transition models are rather complex and produce long titles.

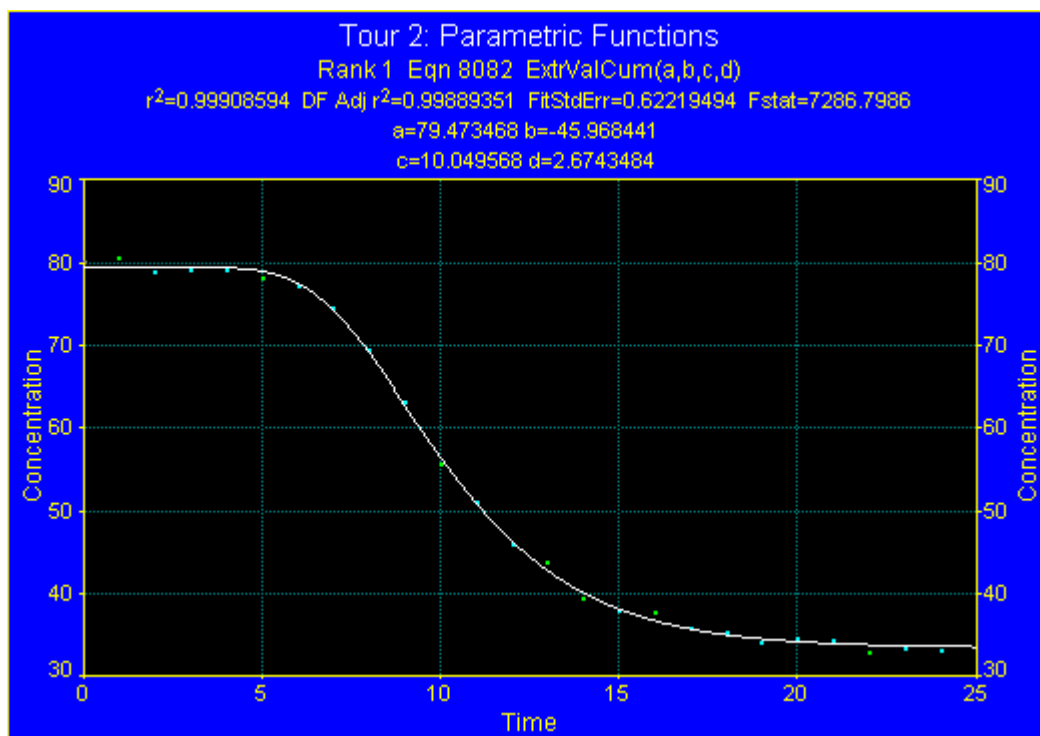
Select the **Preferences** item in the **Curve-Fit** Graph's **File** menu and be sure that the **Notational Format for Non-Linear Equations** option is checked. Press **OK**.

To better see the various goodness of fit indices within the ranked ordering, we will display all four criteria within the **Equation List**.

Using the **List** menu of the **Equation List** window or the **List** menu of the **Curve-Fit** Graph, select the **All Goodness of Fit in List** option.

Extreme Value Cumulative

The highest ranked equation by F-statistic is the four-parameter Extreme Value Cumulative equation. This is clearly a model that fares very well in fitting this data. The higher the F-statistic, the more accurately a given model can be said to describe the data. The F-statistic of this equation is 7287, quite good given the noise in the data.



Note that the third-ranked Asymmetric Sigmoid with Reverse Asymmetry, a 5-parameter equation, offers a higher r^2 and a lower Standard Error.

10 Equations [Rank, r^2 , DOF Adj r^2 , Fit Std Err, F-statistic, FP, Eq#, Eqn]							
File	Edit	List	Filter	Sort			
1	0.9990859	0.9988935	0.6221949	7286.7986	28	8082	ExtrValCum(a,b,c,d)
2	0.9990377	0.9988351	0.6384066	6921.0817	18	8013	LgstcDoseRsp(a,b,c,d)
3	0.9992909	0.9990939	0.5622572	6693.7583	52	8092	AsymSigR(a,b,c,d,e)
4	0.9989608	0.9987420	0.6634325	6408.2846	55	8080	LogNormCum(a,b,c,d)
5	0.9992418	0.9990311	0.5814108	6259.6865	123	8189	EMGCum(a,b,c,d,e)
6	0.9991021	0.9988527	0.6326943	5285.3057	51	8086	PulsePwCum(a,b,c,d,e)
7	0.9986264	0.9983373	0.7627181	4846.8756	17	8084	PulseCum(a,b,c,d)
8	0.9990065	0.9987305	0.6655154	4776.3946	51	8090	AsymSig(a,b,c,d,e)
9	0.9986560	0.9982827	0.7740586	3529.5258	30	8088	WeibullCum(a,b,c,d,e)
10	0.9969114	0.9962611	1.1437293	2151.7753	51	8012	GaussCum(a,b,c,d)

The F-statistic suggests the Extreme Value Cumulative is the superior model since it essentially accomplishes with four parameters what the Asymmetric Sigmoid with Reverse Asymmetry requires five to achieve, although it offers the best least-squares fit in doing so.

Scanning Equations



Use the arrow buttons in the Curve-Fit Graph to inspect the other eight equations in the list.

Scanning can also be done by clicking on the equation desired in the Equation List, by using cursor keys if the Equation List window has the active focus, or by selecting items in the Scan menu of the Curve-Fit

Graph. If the Curve-Fit graph window is active and you have a scroll-wheel mouse, you can scroll the equations using the mouse wheel.

The curve-fit graphs for the asymmetric transition models are largely indistinguishable except for the indication of the undefined condition at 0 for the Logistic Dose Response and Log Normal Cumulative and the sharp initiation of the transition for the Pulse Cumulative and Weibull Cumulative. To more readily see the differences we will lock the Extreme Value Cumulative as a reference.

Fit Reference



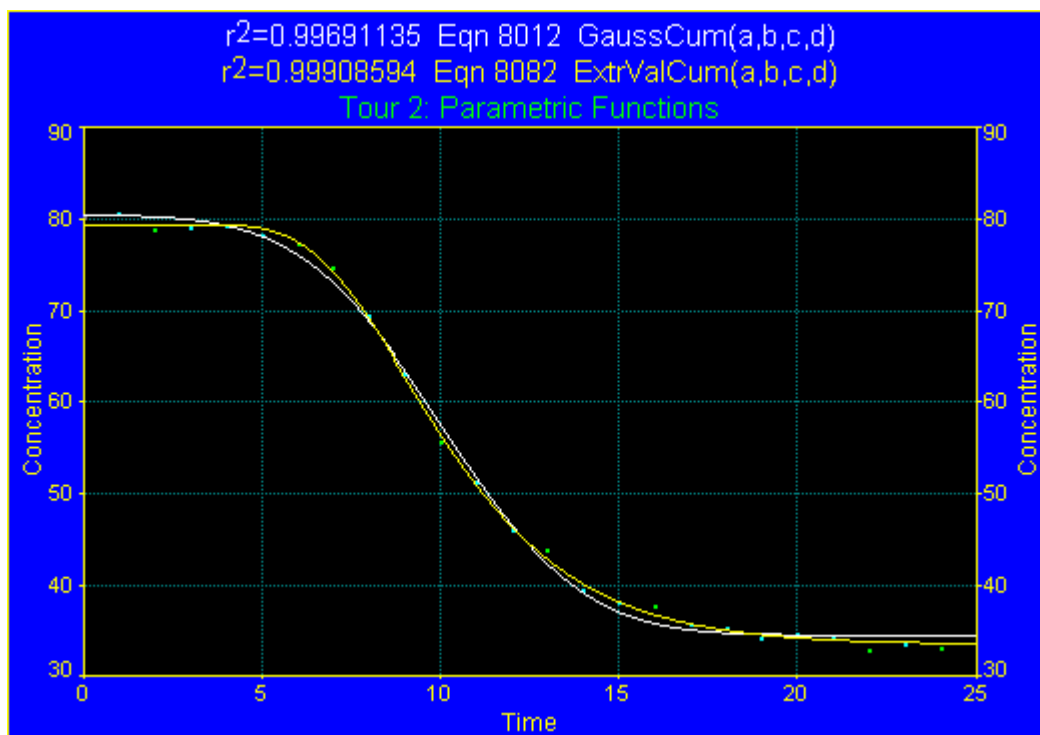
With the Extreme Value Cumulative selected, click the Add Current Equation as Reference button (or the Add Current Equation as Reference in the Reference menu).

Initially, the reference and current curve will be identical, overlapping one another.



Use the arrow buttons to inspect the other equations against relative to this reference.

The subtle differences should be much more apparent.



Use the Toggle Display of Equation References button to disable the display of the references.



Use the double up arrow button to return to the Extreme Value Cumulative equation.

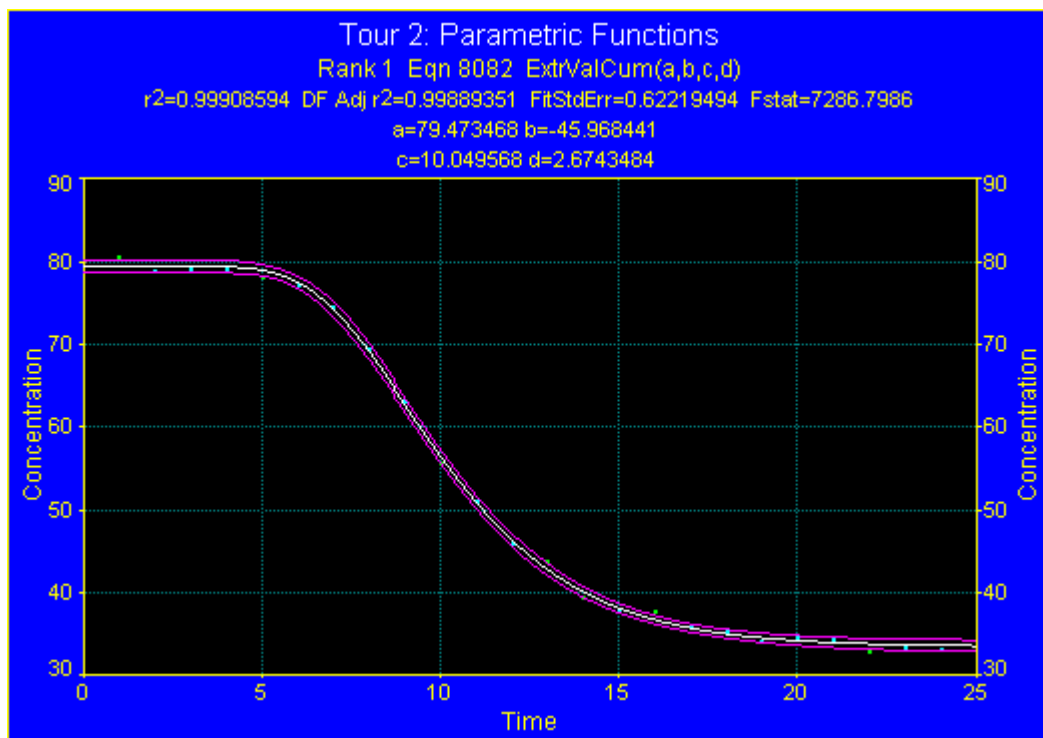
Confidence and Prediction Intervals

The confidence intervals are a measure of how accurately the average curve for repeated experiments is determined. The prediction intervals measure how accurately an individual curve is determined relative to the next experiment's expected values. The prediction bands will always be wider than the confidence

bands. We will assume this data represents the average of a considerable number of trials, and as such, we will use only the confidence intervals.



Click on the Set Confidence/Prediction Intervals, %Confidence button in the graph's toolbar. Be sure the Confidence type of interval is checked, the Prediction type of interval is unchecked, and that the 99% level is checked. Click OK.



Scan through the equations noting the breadth of the confidence intervals for each equation.

Note the widening of the interval at the sharp initiation of the Weibull transition.



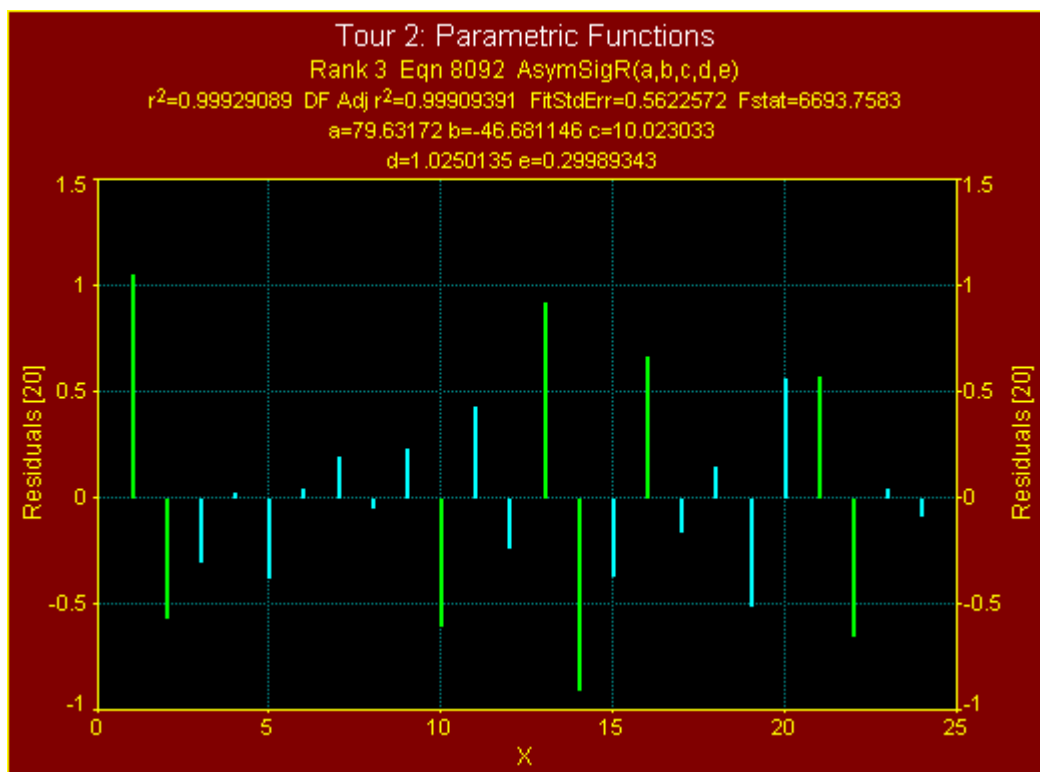
When finished, toggle the intervals off using the Show Confidence/Prediction Intervals button and re-select the Extreme Value Cumulative equation.

Residuals

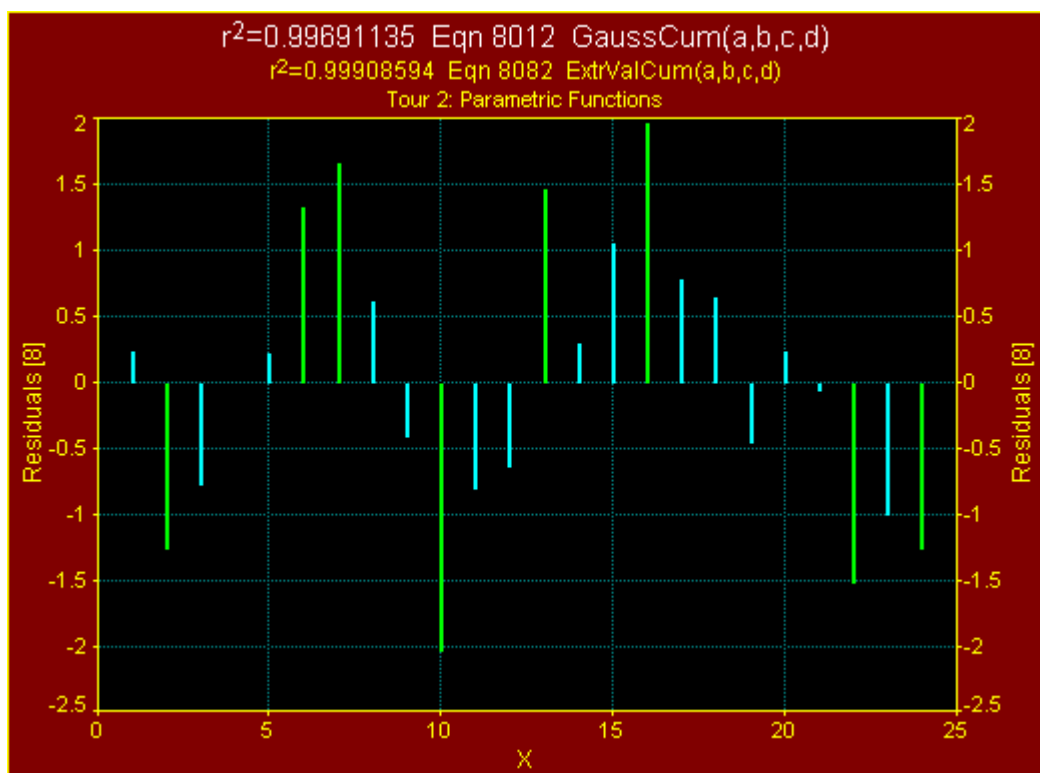


Click on the Resid button to display the residuals for the Extreme Value Cumulative fit. Scan through the equations noting the degree of randomness of the residuals for each.

The number appended to the Y title is the runs count, the number of times the residuals display a sign change across the X range. You will note that while all of the equations produce little in the way of a systematic trend, the runs count is highest for the Asymmetric Sigmoid with the Reverse Asymmetry, the equation with the smallest fit standard error.



When systematic trends are observed in the residuals, it is usually an indication of an inappropriate or incomplete model. Note the clear systematic trend on the Gaussian cumulative, the symmetric transition model we added to the equation set:



The residuals must be normally distributed in order for the standard errors and confidence statistics to be valid.

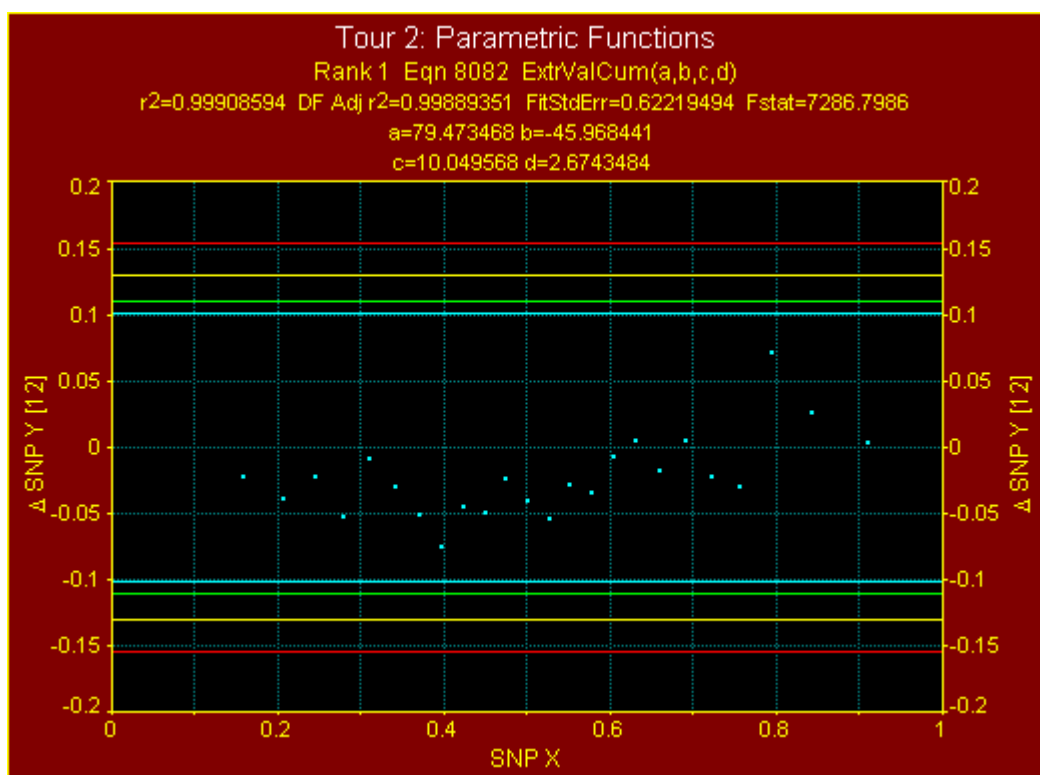
Stabilized Normal Probability Residuals Plot



Using the toolbar in the Residuals Graph, click on the **Display Residuals in Stabilized Normal Probability Plot** button.

Note that the residuals format buttons are also displayed on the left side of the dialog.

On a delta SNP graph, perfectly normal residuals plot as a $y=0$ line and critical limit lines plot as horizontal lines above and below. TableCurve 2D plots 90%, 95%, 99%, and 99.9% critical limit lines. The 99% critical limit means that in only 1 out of 100 data sets should even a single point appear outside the limit.



In this case, all points are easily within even the 90% limit for all of the fitted equations, readily confirming this assumption of Gaussian errors.



Click on the **Display Basic Residuals** button to restore the traditional format.



Close the Residuals window either directly or by again pressing the **Resid** button in the control panel of the Curve-Fit Graph.

If it is not currently selected, click on the top-ranked Extreme Value Cumulative equation in the Equation List.

Numeric Summary

Click on the **Numeric** button to open a numeric summary of the Extreme value Cumulative fit. Scroll to the parameter table.

Rank 1 Eqn 8082 ExtrValCum(a,b,c,d)

r ² Coef Det	DF Adj r ²	Fit Std Err	F-value
0.9990859398	0.9988935060	0.6221949450	7286.7986262

Parm	Value	Std Error	t-value	99% Confidence Limits		P> t
a	79.47346826	0.281909107	281.9116744	78.67134111	80.27559541	0.00000
b	-45.9684411	0.413311267	-111.219908	-47.1444520	-44.7924301	0.00000
c	10.04956768	0.062296479	161.3183907	9.872313038	10.22682232	0.00000
d	2.674348365	0.076175292	35.10781902	2.457603788	2.891092942	0.00000

Area Xmin-Xmax Area Precision

1212.0510118 1.4381e-17

Function min	X-Value	Function max	X-Value
33.677619403	24.000000000	79.473468201	1.0000017963

1st Deriv min	X-Value	1st Deriv max	X-Value
-6.323351375	9.0693820931	-4.679e-07	1.0000015911

2nd Deriv min	X-Value	2nd Deriv max	X-Value
-1.986044726	6.4955283716	1.0355597619	11.643241597

Procedure Minimization Iterations

LevMarqdt Least Squares 7

r ² Coef Det	DF Adj r ²	Fit Std Err	Max Abs Err
0.9990859398	0.9988935060	0.6221949450	1.1885317988

Source	Sum of Squares	DF	Mean Square	F Statistic	P>F
Regr	8462.7396	3	2820.9132	7286.8	0.00000
Error	7.742531	20	0.38712655		
Total	8470.4822	23			

Date	Time	File Source
May 5, 2000	9:31:39 PM	c:\2d5\tour2.sav

The t-values are nothing more than the parameter values divided by their standard errors. These t-values can serve as indicators of the degree of certainty with which the parameters are determined. Here the upper plateau of the transition is determined with the greatest certainty. The steepness or width of the transition was determined with the least certainty. The standard errors and confidence limits are still quite good for pharmacological data.

The **Options** menu offers the means to toggle on and off the different elements of the Numeric Summary.

The Gaussian Quadrature method, with a 1E-8 target precision, is used for the area of the function from Xmin to Xmax. Brent's method is used for the function and derivative extrema. Note that the X value for the first derivative minimum is the center for the underlying Extreme Value peak. All TableCurve 2D transition functions report a true X50, the X value at 50% of the transition. Only for symmetric transitions will the X50 equal the center of the underlying peak.

Scroll the numeric window so that the parameter table is displayed and then scan down two equations (or directly select the Asymmetric Sigmoid Rev equation).

Rank 3 Eqn 8092 AsymSigR(a,b,c,d,e)

r ² Coef Det	DF Adj r ²	Fit Std Err	F-value
0.9992908869	0.9990939110	0.5622571952	6693.7582600

Parm	Value	Std Error	t-value	99% Confidence Limits		P> t
a	79.63171966	0.299006021	266.3214590	78.77628301	80.48715630	0.00000
b	-46.6811462	0.415867559	-112.250031	-47.8709161	-45.4913764	0.00000
c	10.02303344	0.057336511	174.8106630	9.858997440	10.18706945	0.00000
d	1.025013498	0.110559341	9.271161411	0.708710462	1.341316534	0.00000
e	0.299893428	0.046812693	6.406241693	0.165965378	0.433821479	0.00000

Note the smaller t-values and the wider confidence limits, and the same trend as to which parameters are most accurately determined. Here, though, there are two width terms and neither is determined with a great deal of certainty. For the F-statistic to increase with an additional parameter, that parameter must make a significant contribution to improving the overall fit. In this example, there is no evidence to support the use of a five parameter model.

Close the Numeric window and select the Extreme Value Cumulative equation.

Data Summary

The **Data** option displays a point by point data summary with residuals, confidence limits, and prediction limits.

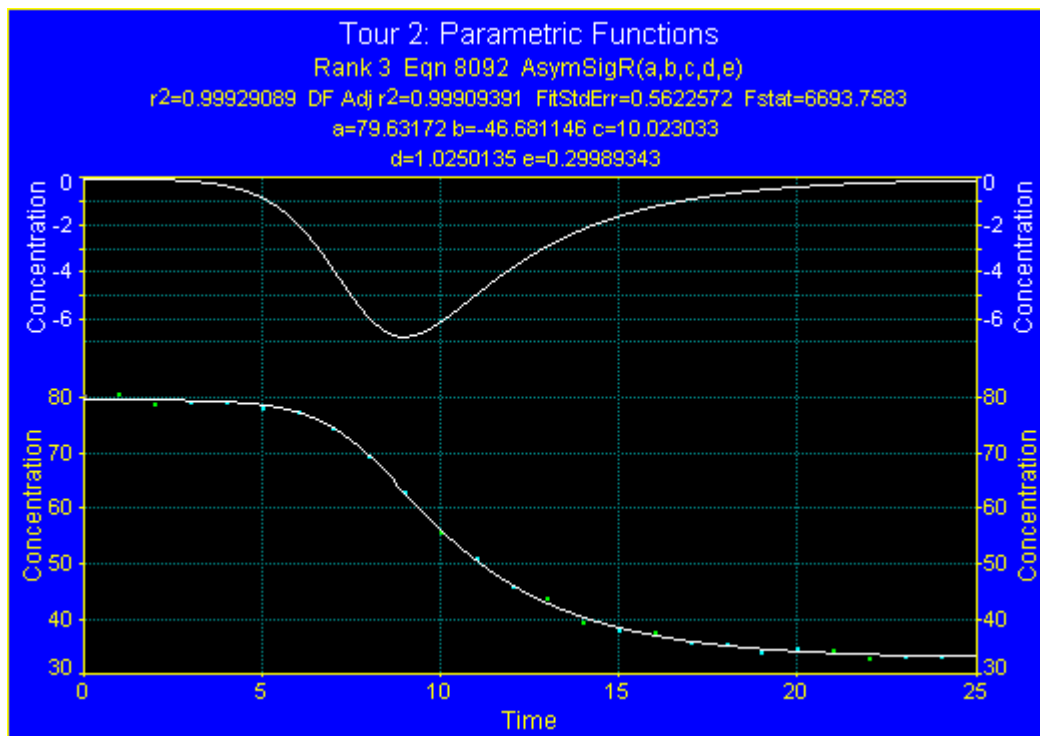


Use the **Data** button in the Curve-Fit graph control panel to open a Data window. Note the information available and then close the window, either directly or by again using the **Data** button.

Derivatives and Cumulative Areas



Click the **Toggle First Derivative on Y2 Axis** in the **Curve-Fit Graph's** control panel.

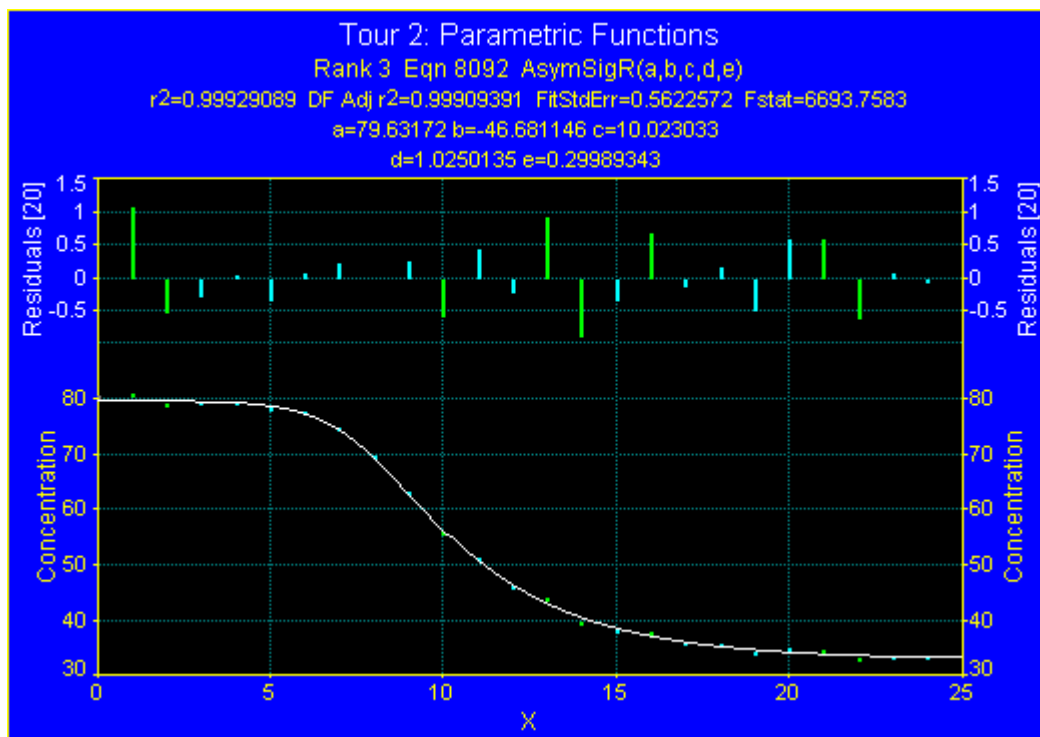


The first derivative curve is an inverted Extreme-Value peak shape since this peak is the derivative of the Extreme-Value Cumulative.

One-to-One Fit and Residuals



To see the Residuals and Fit together, click the **Toggle Residuals on Y2 Axis** button.



Click the **Toggle Residuals on Y2 Axis** button to turn off the Y2 residuals and then click.



Click **OK** to exit the Review.

User-Defined Functions

Let us assume an inventive pharmacologist wanted to see if the models associated with the throughput in a chromatographic column would have any applicability to this biochemical disappearance of this drug in the human bloodstream. Drugs often bind to various molecular sites in ways that could possibly be analogous to the chromatographic mechanisms.

The Exponentially-Modified Gaussian and the Haarhoff-VanderLinde equation are two popular chromatographic peak models. We will use TableCurve 2D's UDF feature to fit cumulatives of these two models to this pharmacological data. TableCurve 2D can store its UDFs in single files, or multiple files can be saved to UDF libraries. Here we will load a library that contains the Cumulative EMG and Cumulative HVL models.

Although TableCurve includes the Cumulative EMG model in its built-in set, we will also fit this same model using a UDF.



Select the **Process** menu's **User Functions** item. In the UDF entry screen, click on the **Read UDF Library** item and then select the UDF file **SAMPLE01.UDL**. Click **OK**.

User Defined Function 1

Read UDF Library Save UDF Library Clear All UDFs Selected UDF: 1

Function Name: Cum EMG Adjustable Parameter Count: 5

Function:

```
RT2_2=SQRT(2)/2
F1=ERF(RT2_2*(X-A2)/A3)
F2=ERF(RT2_2*((X-A2)/A3-A3/A4))
F3=EXP(A3*A3/(2*A4*A4)+(A2-X)/A4)
Y=A0+0.5*A1*(1+F1*(1+F2)*F3)
```

Copy Cut Paste General Trigonometric Statistical Bessel

Clear Current UDF Calculus XY Data Table Non-Lin PeakFit

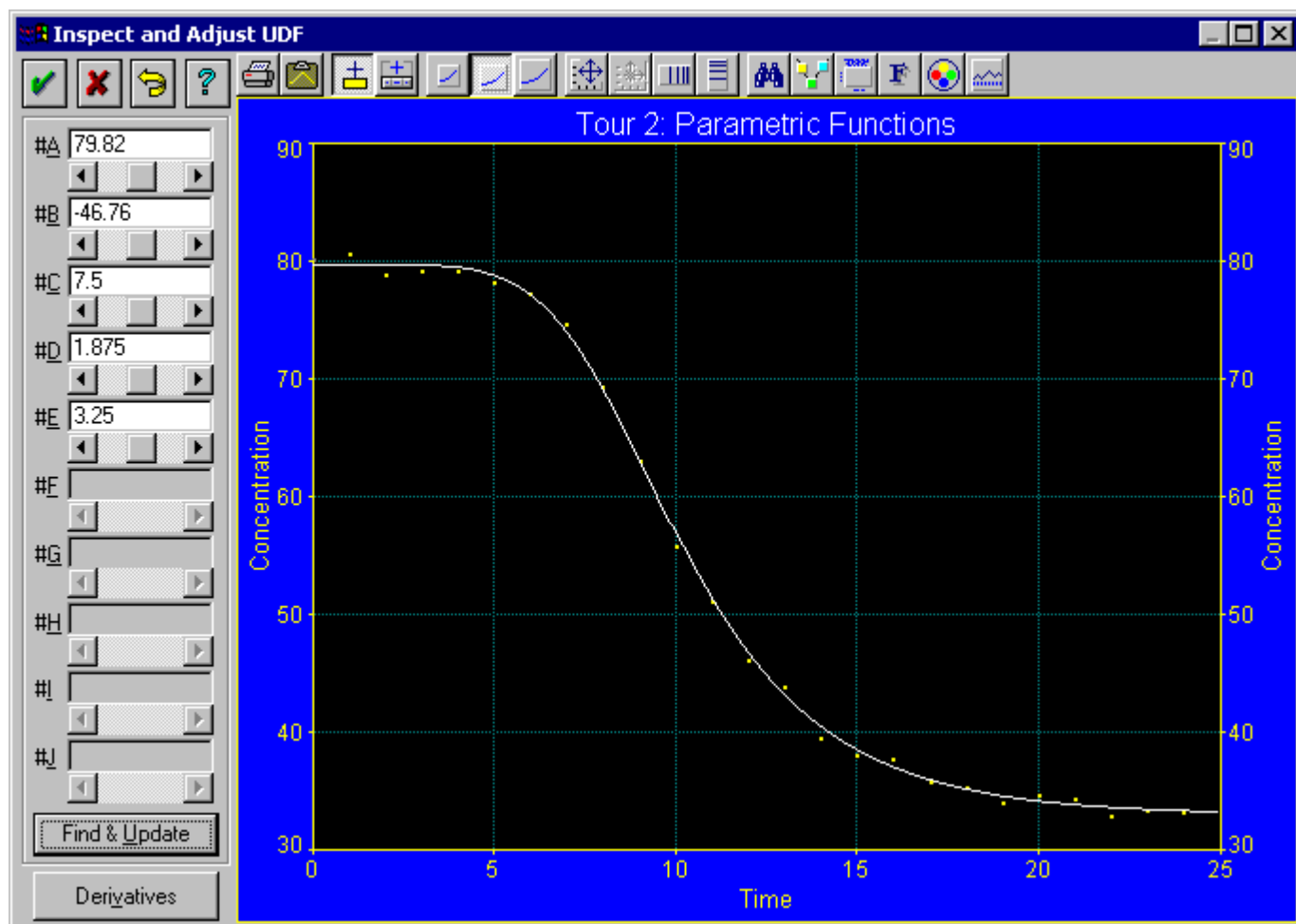
	Min	Start	Max
#A [A0]	-1E+25	YMAX	1E+25
#B [A1]	-1E+25	YATXMAX-YATXMIN	1E+25
#C [A2]	-1E+25	X50	1E+25
#D [A3]	-1E+25	ABS(XWTR)	1E+25
#E [A4]	-1E+25	.5*A3	1E+25
#F [A5]	-1E+25	1	1E+25
#G [A6]	-1E+25	1	1E+25
#H [A7]	-1E+25	1	1E+25
#I [A8]	-1E+25	1	1E+25
#J [A9]	-1E+25	1	1E+25

Copy → Read Save Adjust Fit UDFs OK Cancel Help

The user functions each use A0, A1, A2, etc. for the adjustable parameters. The Cumulative EMG is a rather complex five-parameter model. In this UDF, parameter formulas are used to make the UDF applicable to any range of X and Y data. TableCurve 2D uses specific rules for constructing UDFs. Many of these exist to assist the UDF compiler in producing equations that run nearly as fast as built-in equations.

UDF Auto Adjust

To insure a successful convergence in the non-linear fit, press the **Adjust** item. Rather than adjust individual parameters, use the **Find & Update** item to automatically improve the estimates.



✓ Press **OK** once the estimates have been refined and answer **Yes** to replacing the formulas with the revised numeric values.

So long as UDFs will not be saved back to disk, these modifications are one-time only adjustments.

The 1-15 numeric keypad in the dialog is the control center for selecting from among TableCurve 2D's 15 UDF equations that can be active at any given time. The Cumulative HVL equation was previously saved in position 2 of the UDF library.

Click on the **2** button to change to the second UDF.

There may be a small delay as the first UDF is validated and compiled. Note that the Cumulative HVL is also a complex five parameter transition function.

Again select the **Adjust** item and then the **Find & Update** item to automatically improve the estimates.

✓ Press **OK** once the estimates have been refined and answer **Yes** to replacing the formulas with the revised numeric values.

✓ **OK** From the main UDF entry screen click **OK** to return to the main menu.



Select the Process menu's Curve-Fit Transition Functions.



Press Fit to initiate the fitting of these functions.

When the fit is concluded, enter the Review by clicking the Graph Start button.

All active UDFs are included with any fitting procedure. Note that twelve equations are now added to the list. Note how quickly the two UDFs were fitted.

Since our first UDF is equivalent to the built-in EMG Cumulative (Eqn. 8189), we need merely confirm that it replicates the results from this built-in model. The primary question is whether or not the five parameter HVL cumulative can produce a better F-statistic than the simpler four parameter models.

Note that the Cumulative EMG UDF does indeed exactly match the fit of the built-in function. Also note that the Cumulative HVL UDF is not an effective model by F-statistic.



Click on OK to exit the Review. Exit the program by closing the main window or by the Exit item in the File menu.

The fitting in this example was successful in finding several good models that effectively describe this particular pharmacological data set. The simpler four parameter Extreme Value Cumulative and Logistic Dose Response equations are the most obvious candidates. This type of pharmacological data is traditionally fitted with the Logistic Dose Response model.

Non-Parametric Interpolation

This tutorial covers the non-parametric estimation (interpolation) capabilities of TableCurve 2D.

There are instances where the data trend cannot be accurately accounted by a parametric model. Even the highest order polynomial or rational functions can fail to approximate a complex trend. In these cases, non-parametric procedures are needed for accurate interpolations.

When there are systematic trends within the residuals of even the best parametric models, the equation is insufficient to map the underlying data trend. In this case, one or more of TableCurve 2D's non-parametric estimation methods may be more effective in modeling the trends within the data.

Also, there are certain algorithms within TableCurve 2D that require a constant x-spacing. This is not true for curve-fitting, but is true for all Fourier and Eigen domain procedures. Constant increment x values are also needed for the AR prediction and Savitzky-Golay algorithms. When data lack a uniform x-spacing, using a non-parametric interpolation procedure is often simpler than finding a parametric model upon which to base the interpolations.

Smoothing, Estimation, and Prediction

In TableCurve 2D, a **smoothing** or **denoising** procedure is one where the algorithm always returns smoothed data at the original set of x values. No interpolation or extrapolation is available, and there is no continuous estimation function.

A **prediction** procedure applies to data with equally spaced x-values and generates estimates at additional discrete x values, at one or both ends of a data stream, as in time-series forecasting. No interpolation occurs and there is no continuous function, although it is possible to estimate or predict an arbitrary number of additional data values with this same x-spacing.

An **estimation** procedure, on the other hand, is one where the set of x values used for the computation of y values is arbitrary and user-defined. Interpolation is always possible, and as with curve-fit equations, extrapolation or prediction is available, though not always wise. A continuous function, or the means to simulate such, is the central feature of the procedure.

Parametric and Non-Parametric Procedures

In TableCurve 2D, a **parametric** procedure is one where a set of numeric parameters having intrinsic significance is returned to the user. A **non-parametric** procedure in TableCurve 2D is one where underlying parameters, if present, are typically intermediates and not generally returned to the user.

Strictly speaking, all procedures are parametric in that some parameter set is always computed. Instead of equation coefficients, they may be Fourier coefficients, Savitzky-Golay or Kaiser Bessel filter coefficients, eigenvectors and eigenvalues, local regression coefficients, spline coefficients, and so forth.

Smoothing/Denoising and Estimation Analogs

The **Filter** menu offers procedures that modify the data table by smoothing, removing noise, and isolating components. All of these options return processed data having the original set of x values.

The **Estimate** menu contains all of TableCurve 2D's non-parametric estimation procedures and also its procedure for AR (autoregressive) prediction. Here the original x values are typically modified, extended in the case of AR prediction and set to interpolation bounds and density for the estimation routines.

This tutorial covers only the interpolation available from the **Estimate** menu.

- Smoothing and Denoising is covered in the [Noise Reduction](#) tutorial.
- Component Isolation via Eigendecomposition is covered in the [Eigendecomposition Filtering](#) tutorial.
- Component Isolation via Fourier decomposition is covered in the [Fourier Filtering](#) tutorial.
- Prediction is covered in the [Prediction and Forecasting](#) tutorial.

Estimation Procedures



The [Spline Estimation](#) option offers eight important spline estimation procedures. Five of these are smoothing splines: a cross-validation cubic, NURBS, and least-squares B-splines.



The [Fourier Estimation](#) option is a Fourier filtering procedure with interpolation and smoothed estimations based upon the evaluation of the individual Fourier components.



The [Smoothed Data Spline Estimation](#) procedure combines automated smoothing and B-spline estimation. These are not smoothing splines, but rather the fitting of an interpolating B-spline to data that have been pre-smoothed.



The [Local Regression Spline Estimation](#) option offers a much greater control of the Loess algorithm, including higher orders, and also adds interpolation.



The [Savitzky-Golay Spline Estimation](#) option additionally adds interpolation with a constrained cubic spline interpolant. This procedure has been especially tailored for estimating derivatives.

Non-Parametric Interpolation

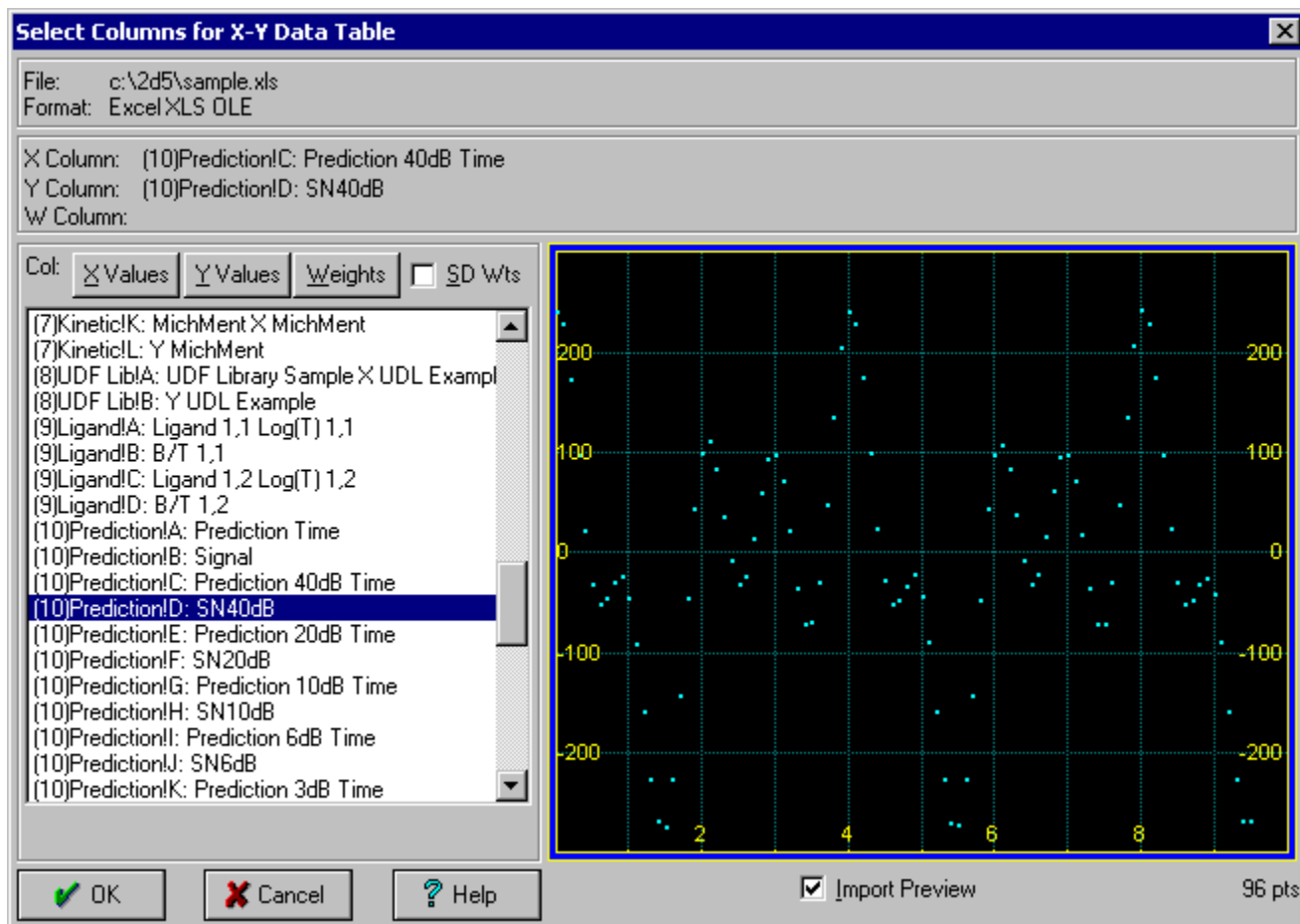
We will begin by exploring the procedures useful for non-parametric interpolation. Although many of the algorithms within these procedures offer smoothing, we will focus primarily on basic interpolation.

Start TableCurve 2D. Select [Start/Programs/TableCurve 2D v5](#).



Select the File menu's [Import](#) option (or use the [Import](#) button in the main toolbar). If [Excel \[xls\]](#) files are not shown, click on the [Files of Type](#) drop-down button and select [Excel \[xls\]](#) files. Select and open the file [SAMPLE.XLS](#).

Select the column with the label [\(10\)Prediction!C: Prediction 40dB Time](#) to be used as the X-variable in the data table. Select the next column identified in the selection list as [\(10\)Prediction!D: SN40dB](#) for the Y-variable. Check [Import Preview](#) to see a graph of the data that will be imported.



Note that the first two selections are automatically placed in the X and Y positions. You may double click on the column for the Y variable in order to immediately proceed with the read operation. You can also revise the initial X,Y selections as well as to specify a column to be used for the weights. The weights can be optionally imported as standard deviations.

Press OK to accept these choices. Press OK once again within the titles dialog to confirm the imported titles.

This data set consists of the sum of three different sinusoids and 1% Gaussian noise.

Parametric Fitting



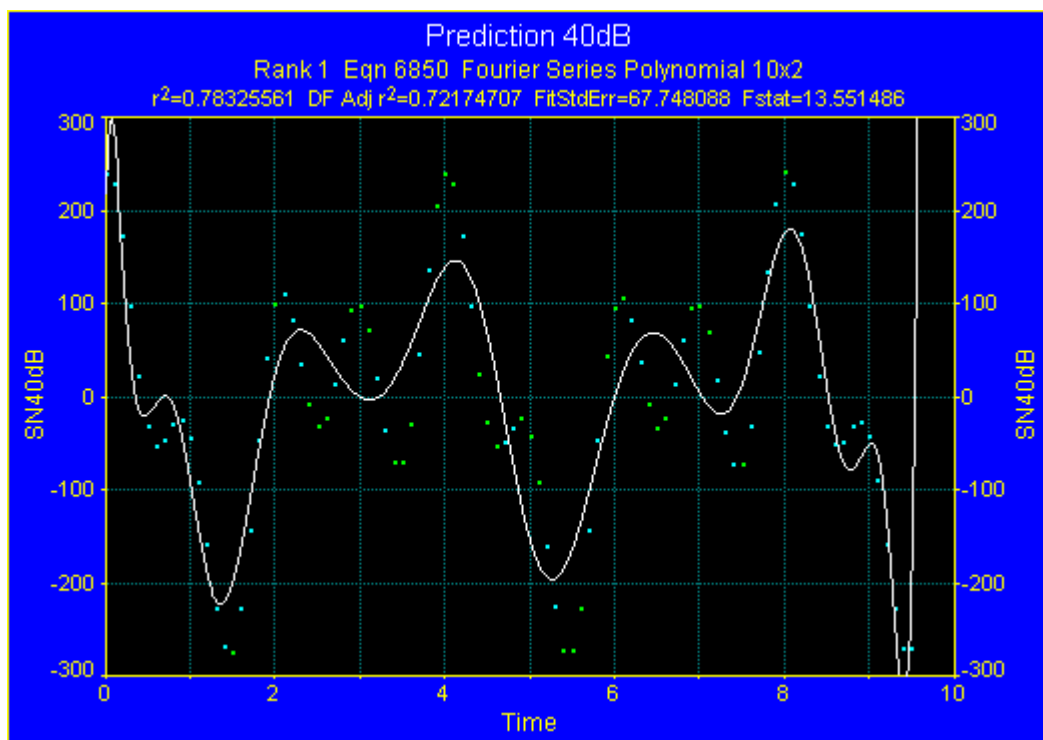
Select the Curve Fit Poly/Ratl option in the Process menu or toolbar.

When the fit is concluded, enter the Review by clicking the Graph Start button.



As an alternative, you may press **OK** and select the **Graph Start** item in the Review menu. If you do not respond to the conclusion of the fit in 10 seconds, the Review is automatically started.

Using the List menu of the Curve-Fit graph, or the Sort menu of the Equation List, select the Sort by r^2 item.



The highest ranked equation, a 21 parameter Fourier series polynomial is unable to map the local minima and maxima in this data set. This is one case where a non-parametric procedure is particularly valuable.

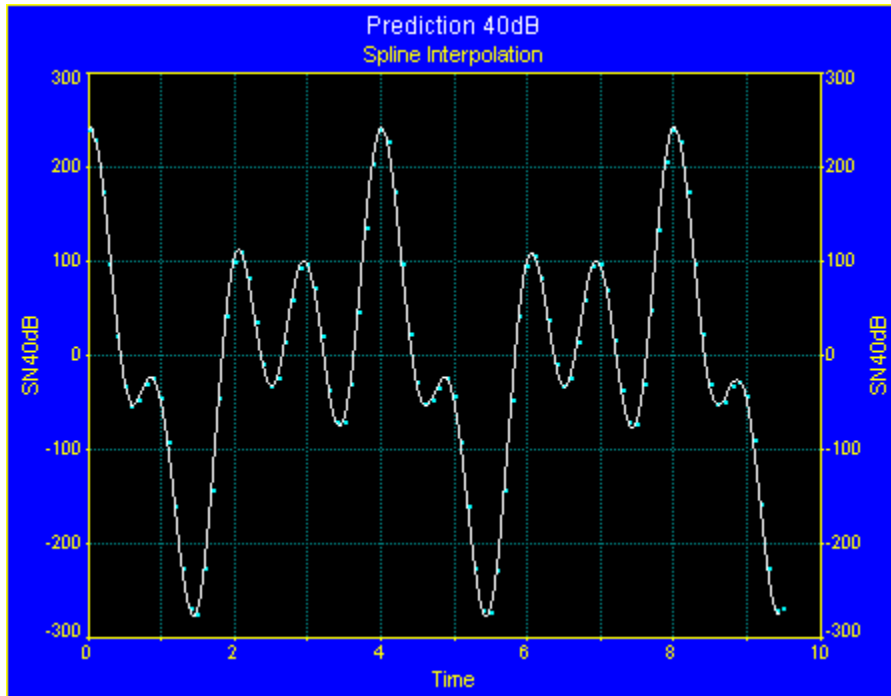


Click OK to close the Review.

Spline Estimation



Select the Spline Estimation option from the Process toolbar or Estimate menu. Select the Cubic algorithm.



The basic cubic spline does an excellent job interpolating the data.

TableCurve 2D offers a broad selection of spline algorithms. The most common types of splines are cubic splines. These produce smooth first derivatives and continuous (but not smooth) second derivatives.

- The **Cubic** option fits a cubic spline interpolant where the breakpoints or knots of the spline are the X values of the data. As such, the knots and order are fixed. The spline passes exactly through each data point.
- The **Cubic constrnd** option is similar, except that it minimizes the 'wiggles' common to cubic splines. The wild swings observed when noisy data are fit to cubic splines do not occur with this algorithm. Again, the spline passes exactly through each data point.
- The **Cubic smooth** option fits a smooth cubic spline to the data using a automatic smoothing procedure known as cross-validation. Since the interpolant will not pass exactly through the data, TableCurve 2D reports a standard error (SE) and r^2 for this type of spline.

There are four B-Spline procedures, three of which offer overdetermined least-squares fits.

- The **B-Spline** option fits the B-spline of a specified order using a knot sequence based upon the X-values of the data. The knots are thus fixed. Orders range from 2 through order 8. The B-spline passes exactly through each data point. The B-Spline of order 3 is identical to the cubic spline.
- The **LS B-Spline fix** option offers a B-spline where the knots are not located at the X-values, but fixed in position, and lower in count than the number of data points. For this option, you must set both the order and the number of knots desired. The smaller the number of knots, the greater will be the smoothing of noisy data. TableCurve 2D reports a standard error (SE) and r^2 for this type of spline.
- The **LS B-Spline opt** option is identical to the previous spline, except that the knots are now variable, and are shifted to further minimize the overall least squares fit. Due to its iterative nature, this algorithm is appreciably slower than all of the other spline options.
- The **LS B-Spline usr** option offers a B-spline where the knots are located at user-specified X-values. For this option to be available, a knot sequence must first be imported. You must also specify the spline order.

There is also a NURBS algorithm.

- The **NURBS** option fits a non-uniform rational B-spline to the data. The knots are fixed at the X values of the data points. The order can vary from 2 to 8. Unlike the wild oscillations sometimes observed with the higher order B-splines, NURBS become smoother with increasing order. A NURBS by its nature is a smoothing spline, and does not pass exactly through the data points. A standard error (SE) and r^2 is thus reported.

Although these algorithms suffice for most estimations, TableCurve offers four other estimation procedures. In most cases, you will use one of these other procedures only because the smoothing offered by the above algorithms is inadequate:

- The best smoothing splines are the least-squares B-splines, but fixed knots do not usually work well, the non-linear optimization that seeks to find the optimum knot sequence may fail, and it can be a tedious process to manually specify a knot sequence.
- The cross-validation smoothing spline is not useful for derivatives beyond first order.
- The smoothing offered by NURBS is generally of limited value.



The **Smoothed Data Spline Estimation** procedure combines TableCurve 2D's automated smoothing with B-spline estimation. The data are pre-smoothed and then fitted to the B-spline. Frequency and eigen domain methods are often superior for removing noise without distorting the underlying deterministic trend.



The **Local Regression Spline Estimation** option offers a much greater control of the Loess smoothing algorithm. Loess is often very effective when the noise is abysmally high and a data trend is not visually perceptible.



The **Savitzky-Golay Spline Estimation** option offers a much greater control of the Savitzky-Golay smoothing algorithm and offers interpolation using the constrained cubic spline. This procedure is recommended for estimating derivatives.



The **Fourier Estimation** option offers Fourier domain filtering with interpolation and smoothed estimations based upon the evaluation of the individual Fourier components.

For more information on the smoothing estimation procedures, please explore the [Noise Reduction](#) tutorial.



Since we are not interested in updating the data table, click on **Cancel to exit the spline estimation procedure.**



Select the **Fourier Estimation option in the Estimate menu or Process toolbar.**

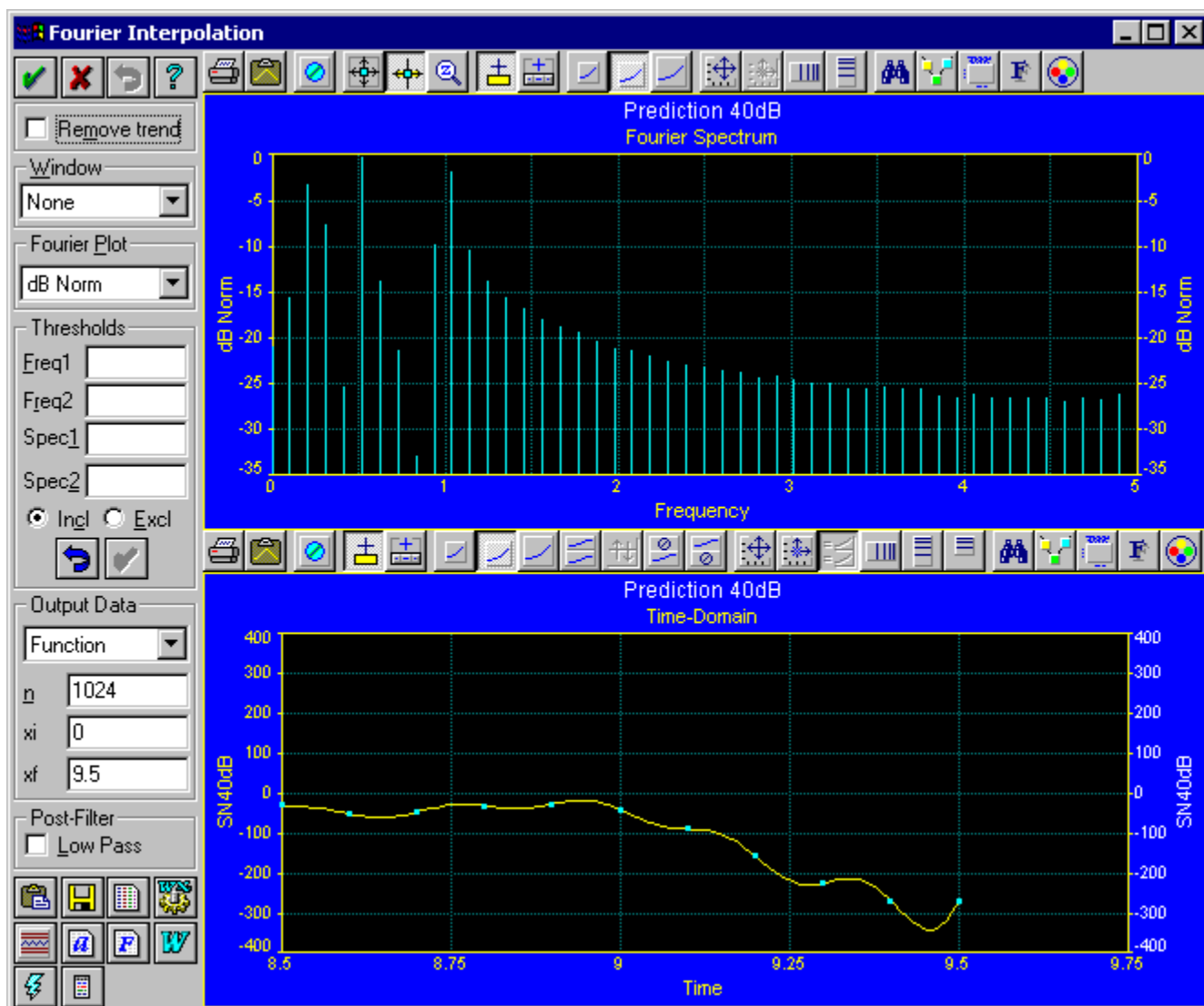
Unless some form of filtering takes place, all settings will exactly reproduce the input data. The quality of the interpolation, however, will vary.

Be sure **Remove Trend is unchecked, set the Window to **None**, be sure **Output Data** is set to **Function**, and change **n** to **1024**.**

In the lower graph, use the mouse to zoom-in the region of data between **x=8.5 and **x=9.5**.**

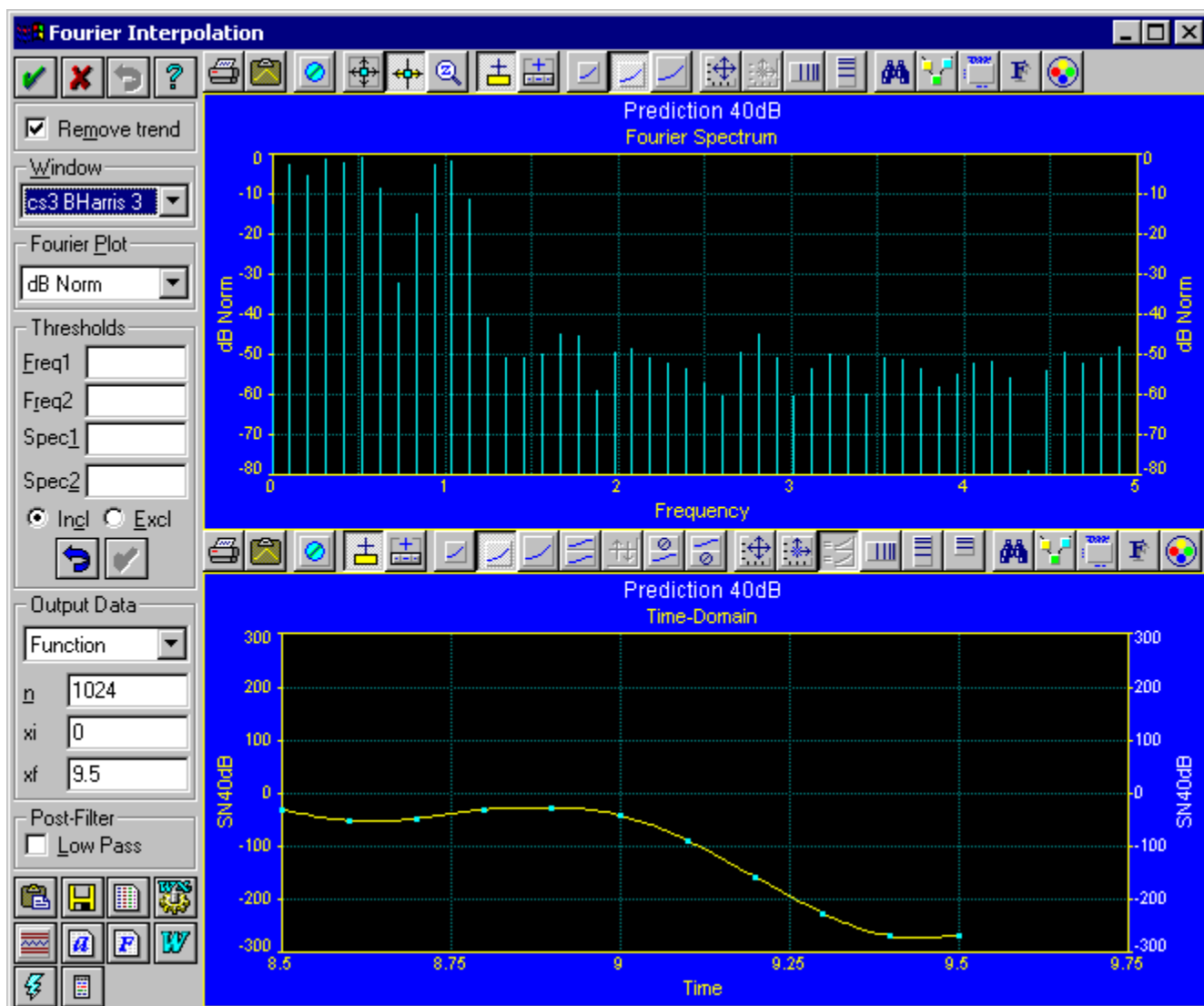


Click on the **Modify Point Format button in the lower graph's toolbar. Select the **Y2 Axis** and set the size to **4**. Click OK to close the dialog.**



Note the oscillations between the last few data values.

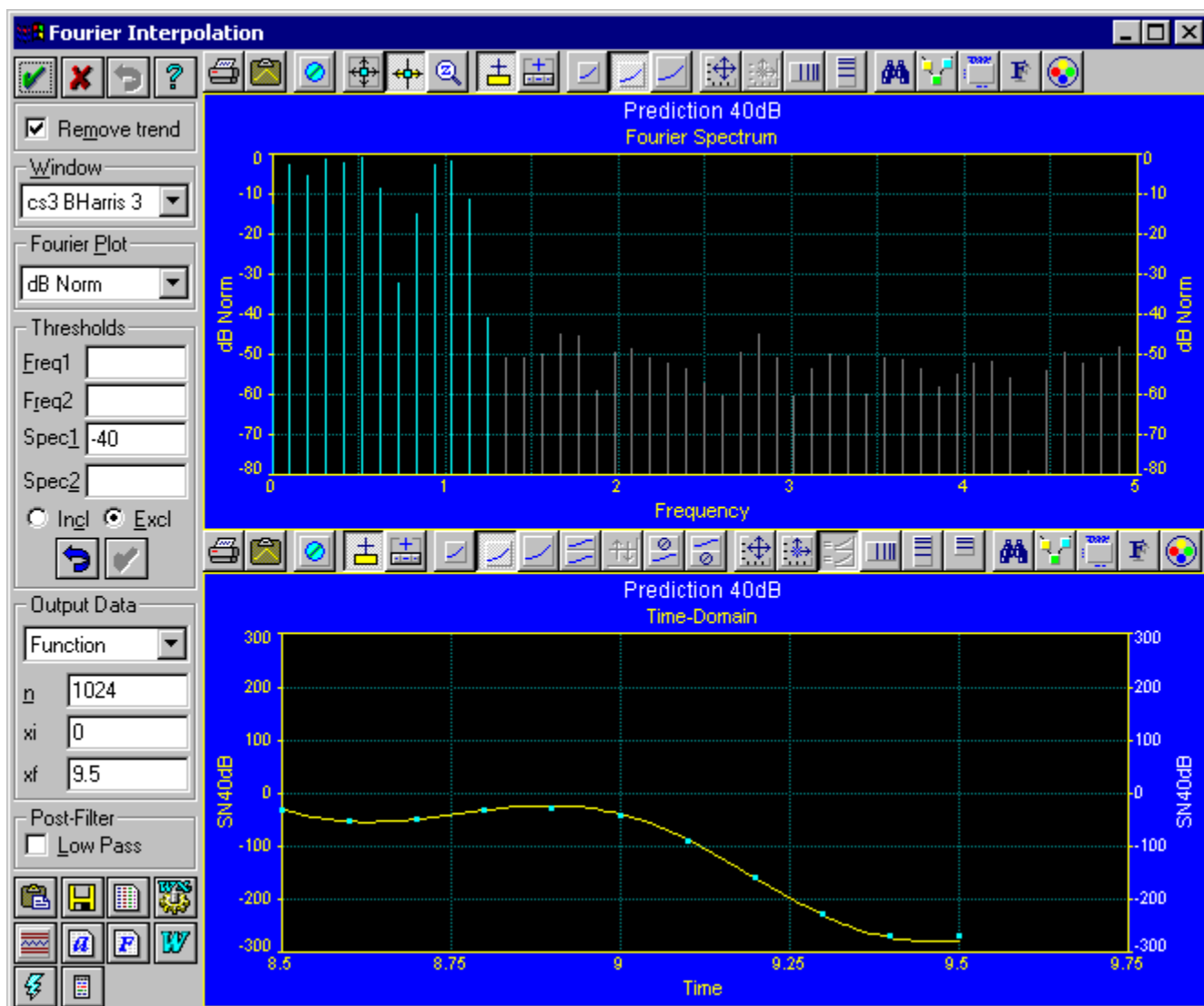
Check the **Remove trend** box and set the **Window** to **cs3 BHarris 3**.



Even though both Fourier decompositions exactly reproduce the input data, they are very different and also produce very different interpolations. The oscillations between the final points are no longer present.

When there is a good indication of a signal-noise threshold, as in this case, it is a simple matter to remove the noise.

Enter -40 for Spec1 and check Excl.



While this removal of the noise may have improved the estimates, note that interpolation is no longer occurring. The Fourier model no longer passes exactly through each data point.

X Since we are not interested in updating the data table, click on **Cancel** to exit the Fourier estimation procedure.

Exit the program by closing the main window or by the **Exit** item in the File menu.

Noise Reduction

This tutorial covers the denoising and smoothing capabilities of TableCurve 2D.

Denoising/Smoothing vs. Curve-Fitting

When fitting a theoretically appropriate parametric model to the trend within a data set, it is not usually desirable to smooth or remove noise prior to fitting. In this particular instance, curve-fitting represents the very finest form of smoothing. As long as the residuals are normally distributed (follow a Gaussian distribution) and lack a systematic trend, the parametric model is a perfectly acceptable way to factor out the noise within a data set. Since a parametric model is also a continuous function, smooth data can be reconstructed at exactly the x values and density desired.

There are instances, however, where the data trend cannot be accurately accounted by a parametric model with a modest number of parameters. Instead, it may be necessary to fit a higher order polynomial or rational function to approximate a complex trend. Here, there are a number of advantages in removing noise prior to fitting. In general, higher order polynomials and rationals tend to fit local trends within the noise. Further, significant noise often results in rational fits with poles (undefined regions) in the x-range of the data. By removing this noise prior to fitting, higher order polynomial and rational fits can be markedly improved.

When there are systematic trends within the residuals of even the best parametric models, the model is insufficient to map the underlying data trend. In this case, one or more of TableCurve 2D's smoothing methods may be more effective in filtering out the noise while minimally impacting the deterministic part of the signal. The Fourier and Eigen domain denoising procedures are often very effective, and unlike parametric fitting, the expertise needed to use these procedures optimally is quite easy to gain.

Smoothing/Denoising, Estimation, and Prediction

In TableCurve 2D, a **smoothing** or **denoising** procedure is one where the algorithm always returns smoothed data at the original set of x values. No interpolation or extrapolation is available, and there is no continuous estimation function.

A **prediction** procedure applies to data with equally spaced x-values and generates estimates at additional discrete x values, at one or both ends of a data stream, as in time-series forecasting. No interpolation occurs and there is no continuous function, although it is possible to estimate or predict an arbitrary number of additional data values with this same x-spacing.

An **estimation** procedure, on the other hand, is one where the set of x values used for the computation of y values is arbitrary and user-defined. Interpolation is always possible, and as with curve-fit equations, extrapolation or prediction is available, though not always wise. A continuous function, or the means to simulate such, is the central feature of the procedure.

Parametric and Non-Parametric Procedures

In TableCurve 2D, a **parametric** procedure is one where a set of numeric parameters having intrinsic significance is returned to the user. A **non-parametric** procedure in TableCurve 2D is one where underlying parameters, if present, are typically intermediates and not generally returned to the user.

Strictly speaking, all procedures are parametric in that some parameter set is always computed. Instead of equation coefficients, they may be Fourier coefficients, Savitzky-Golay or Kaiser Bessel filter coefficients, eigenvectors and eigenvalues, local regression coefficients, spline coefficients, and so forth.

Smoothing/Denoising and Estimation Analogs

The **Filter** menu offers procedures that modify the data table by smoothing, removing noise, and isolating components. All of these options return processed data having the original set of x values.

The **Estimate** menu contains all of TableCurve 2D's non-parametric estimation procedures and also its procedure for AR (autoregressive) prediction. Here the original x values are typically modified, extended in the case of AR prediction and set to interpolation bounds and density for the estimation routines.

Automated Procedures



The **Automated Smoothing** option in the **Filter** menu simplifies the use of six different procedures for smoothing the data table. There include Fourier filtering, Loess, Gaussian convolution, Savitzky-Golay, Eigen filtering, and Kaiser-Bessel filtering. No knowledge of the algorithms is needed and uniform x data are automatically generated when required by an algorithm.



The **Smoothed Data Spline Estimation** procedure in the **Estimate** menu combines automated smoothing and B-spline estimation. These are not smoothing splines, but rather the fitting of an interpolating B-spline to data that have been pre-smoothed.

With only a single smoothing level adjustment, these automated procedures will usually offer a close to optimum level of noise removal for each of the algorithms. In general, distortion of the underlying data trends does not occur until high smoothing levels are specified. In this tutorial, we will begin with the automated procedure. In many instances, its performance will be completely satisfactory.

Specialized Filtering and Estimation Procedures

The more specialized procedures are as follows:



The separate **Savitzky-Golay Smoothing** procedure offers much greater control of the Savitzky-Golay algorithm and smooth derivatives through order eight based directly on the smoothing filter.



The **Savitzky-Golay Spline Estimation** option additionally adds interpolation with a constrained cubic spline interpolant. This procedure has been especially tailored for estimating derivatives.



The **Local Regression Spline Estimation** option offers a much greater control of the Loess algorithm, including higher orders, and also adds interpolation.



The **Fourier Denoising** option offers both frequency and signal thresholding for low pass frequency-domain filtration as well as raised data tapers to reduce spectral leakage.



The **Fourier Filtering** option additionally offers channel by channel control of the Fourier domain filtering. This routine is generally used for component isolation, although it can be used for lowpass denoising.



The **Fourier Estimation** option is a Fourier filtering procedure with interpolation and smoothed estimations based upon the evaluation of the individual Fourier components.



The **Eigendecomposition Denoising** option offers a greater control of the algorithm and the means to visually separate the signal and noise eigenmodes via singular value magnitudes.



The **Eigendecomposition Filtering** option additionally offers the means to visualize individual or pairs

of eigenmodes both in the time and frequency domains. This routine is also typically used for component isolation, although it can also perform signal-noise thresholding.



The [Spline Estimation](#) option offers eight important spline estimation procedures. Five of these are smoothing splines, a cross-validation cubic, NURBS, and least-squares B-splines.

Prediction

TableCurve 2D offers a single prediction procedure:



The [AR Modeling and Prediction](#) procedure offers effective autoregressive forecasting and extrapolation. The AR algorithms include SVD (singular value decomposition) procedures for in-place noise removal. Multiple orders can be simultaneously plotted and stabilizations are available for roots that lie outside the unit circle. The points that are to be processed can be specified, allowing predictions based on a data segment to be compared with subsequent data. The extent of the prediction is variable and white noise can be added to test the robustness of an algorithm's prediction.

Noise Removal

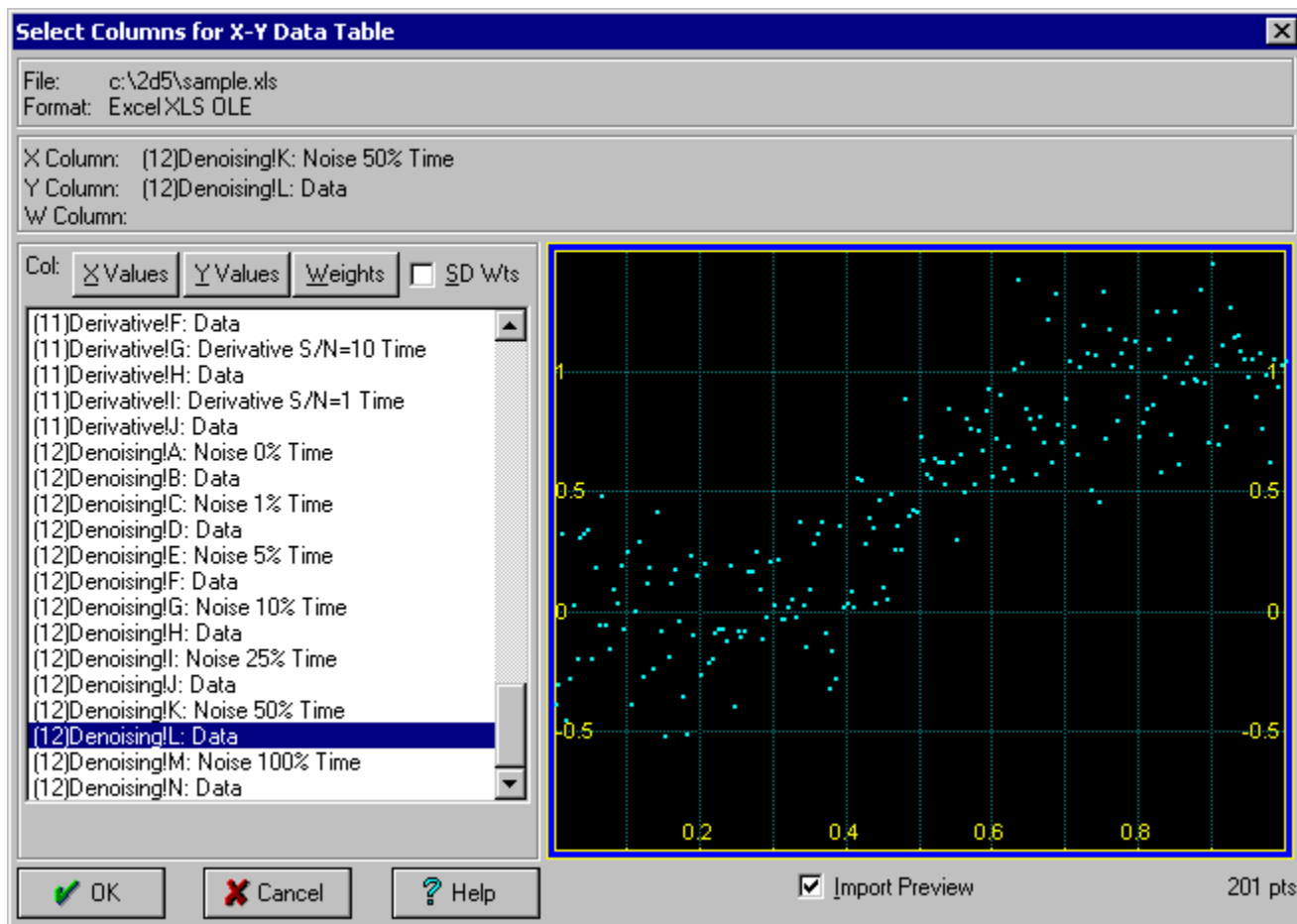
This tutorial covers the filtering of noise from a data set.

Start TableCurve 2D. Select [Start/Programs/TableCurve 2D v5](#).



Select the File menu's [Import](#) option (or use the [Import](#) button in the main toolbar). If Excel [xls] files are not shown, click on the [Files of Type](#) drop-down button and select [Excel \[xls\] files](#). Select and open the file [SAMPLE.XLS](#).

Select the column with the label [\(12\)Denoising!K: Noise 50% Time to be used as the X-variable in the data table](#). Select the next column identified in the selection list as [\(12\)Denoising!L: Data for the Y-variable](#). Check [Import Preview](#) to see a graph of the data that will be imported.



Note that the first two selections are automatically placed in the X and Y positions. You may double click on the column for the Y variable in order to immediately proceed with the read operation. You can also revise the initial X,Y selections as well as to specify a column to be used for the weights. The weights can be optionally imported as standard deviations.

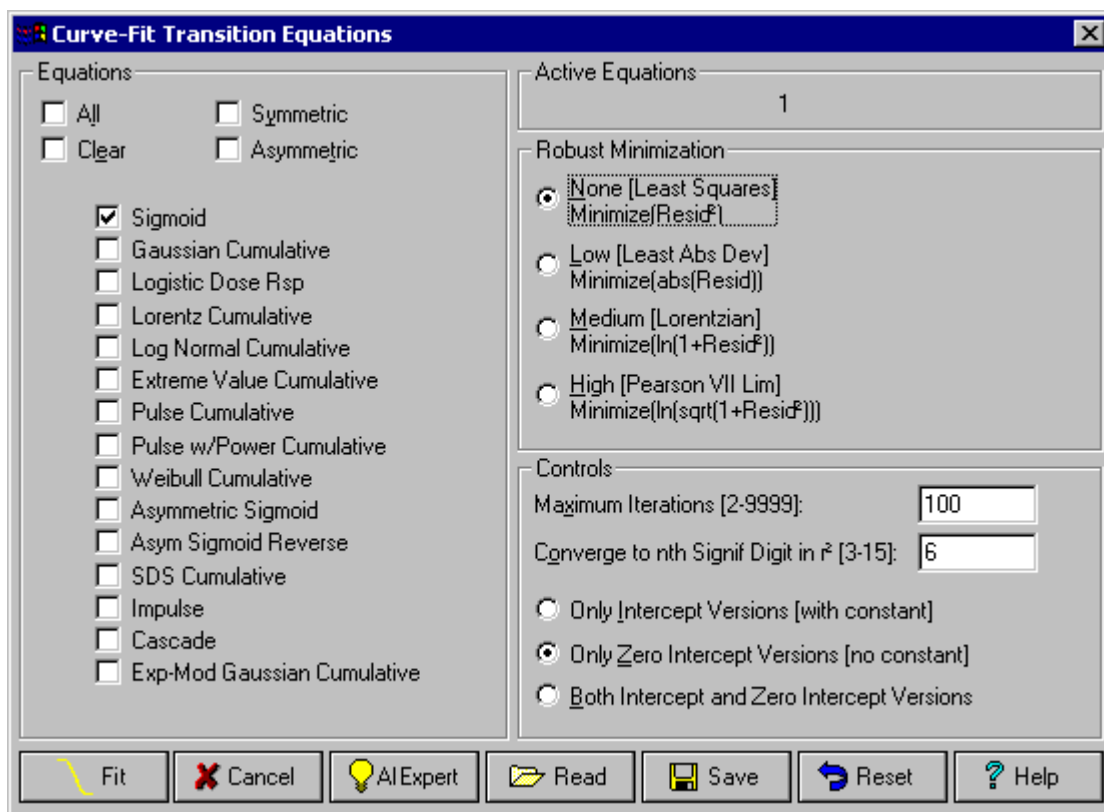
Press OK to accept these choices. Press OK once again within the titles dialog to confirm the imported titles.

This data was generated from a sigmoid model with (1,0.5,0.075) parameters. 50% Gaussian (white) noise was added. This means that the SD of the noise is half the SD of the clean signal.

Parametric Fitting



Select the Process menu's Curve Fit Transition Functions. Click on Clear. Select Sigmoid. Select the Only Zero Intercept Versions (no constant) control.



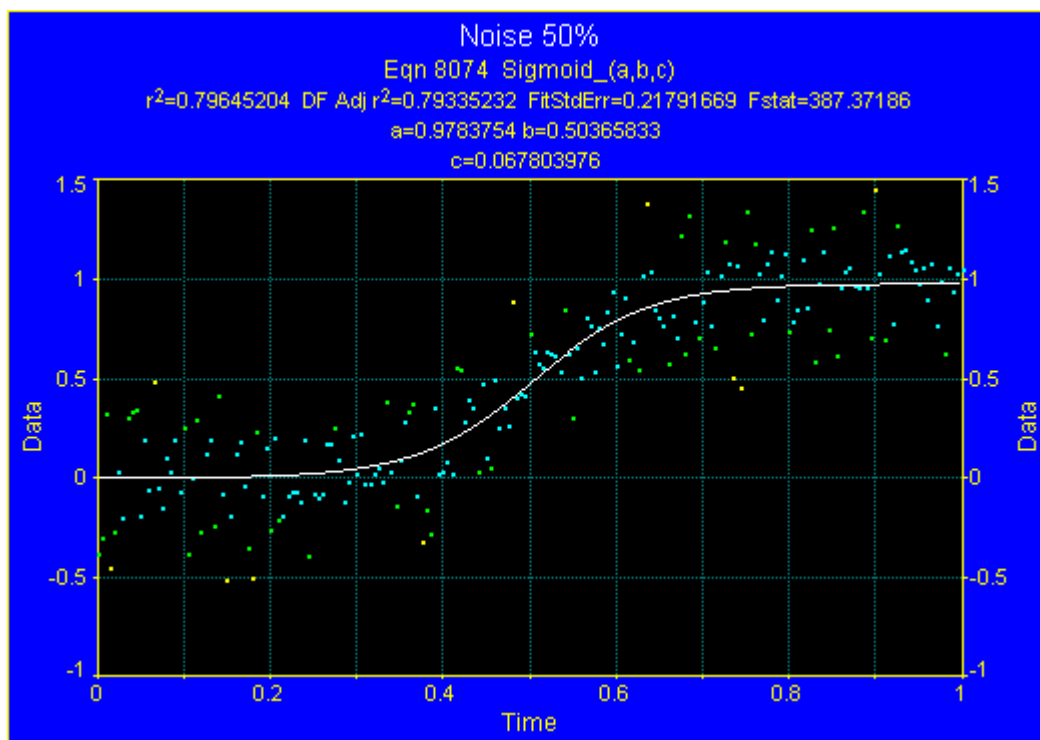
Click the **Fit** button.

When the fit is concluded, enter the Review by clicking the **Graph Start** button.



As an alternative, you may press **OK** and select the **Graph Start** item in the Review menu. If you do not respond to the conclusion of the fit in 10 seconds, the Review is automatically started.

Parametric Smoothing



The Curve-Fit graph reveals a very good recovery of the coefficients. The amplitude of 0.978 compares favorably to the 1.0 in the noise-free data. The transition center of 0.504 is very close to 0.5, and the transition width of 0.068 is somewhat sharper than the 0.075 used to generate the data. The errors are 2.2%, 0.72%, and 9.6%.

Marking Outliers and Refitting

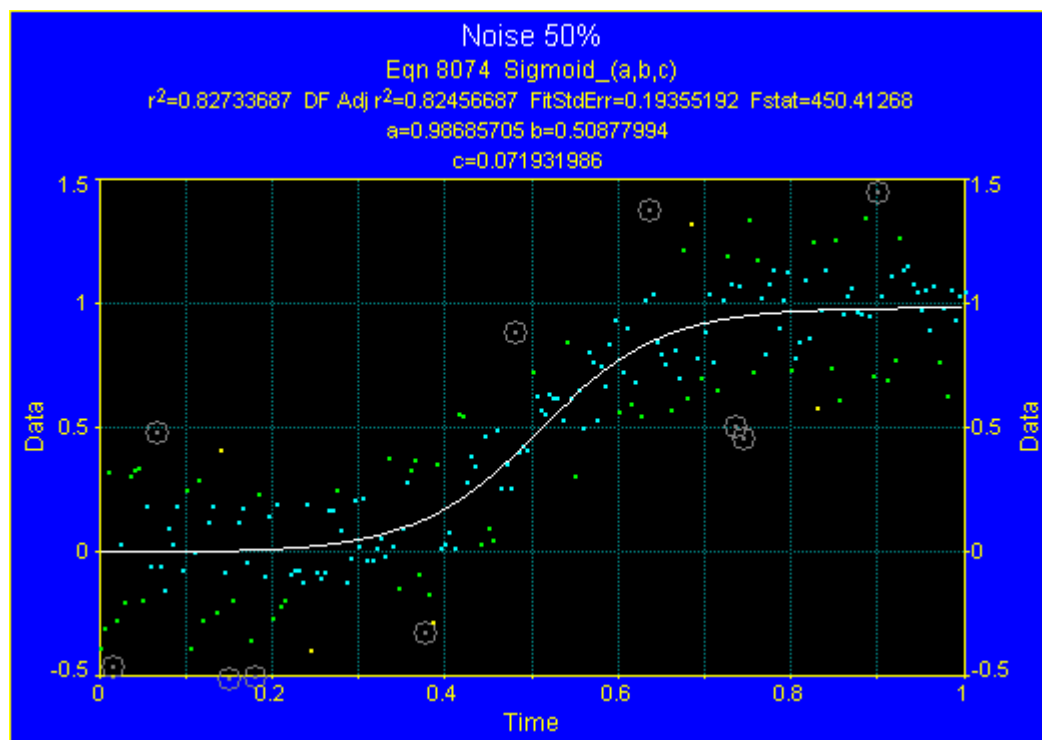


Click on the **Mark Excluded Points for Refit** option. Select **All Points Outside 2 Standard Errors** and click **OK**. Acknowledge that 10 points will be excluded.

The marked points will be toggled in state (in this case disabled or turned off) when a refit is made.



Click on the **Refit Transition Equations and UDFs** button just to the left.



By removing the largest outliers from the fit, the coefficient recovery is improved. The coefficients of 0.987, 0.509, and 0.072 represent errors of 1.3, 1.7, and 4.1% from the noise-free values.



To reactivate all points, once again click on the **Refit Transition Equations and UDFs** button.



Click **OK** to close the Review.

Automated Smoothing



Select the **Automated Smoothing** option from the toolbar or Filter menu.

With the exception of the Loess procedure, all of the algorithms in the **Automated Smoothing** option require uniformly spaced data. If a constant x spacing is not present, this procedure automatically generates uniform data for all of the algorithms except Loess. A constrained cubic spline interpolant is used. This is not a factor for this data set since uniform x spacing is present.

Select the **FFT Filtering** algorithm and click on the **AI Expert** button.

Select the **Loess** algorithm and click on the **AI Expert** button.

Select the **Savitzky-Golay** algorithm and click on the **AI Expert** button.

Select the **Gauss Convolution** algorithm and click on the **AI Expert** button.

Select the **Eigendecomposition** algorithm and click on the **AI Expert** button.

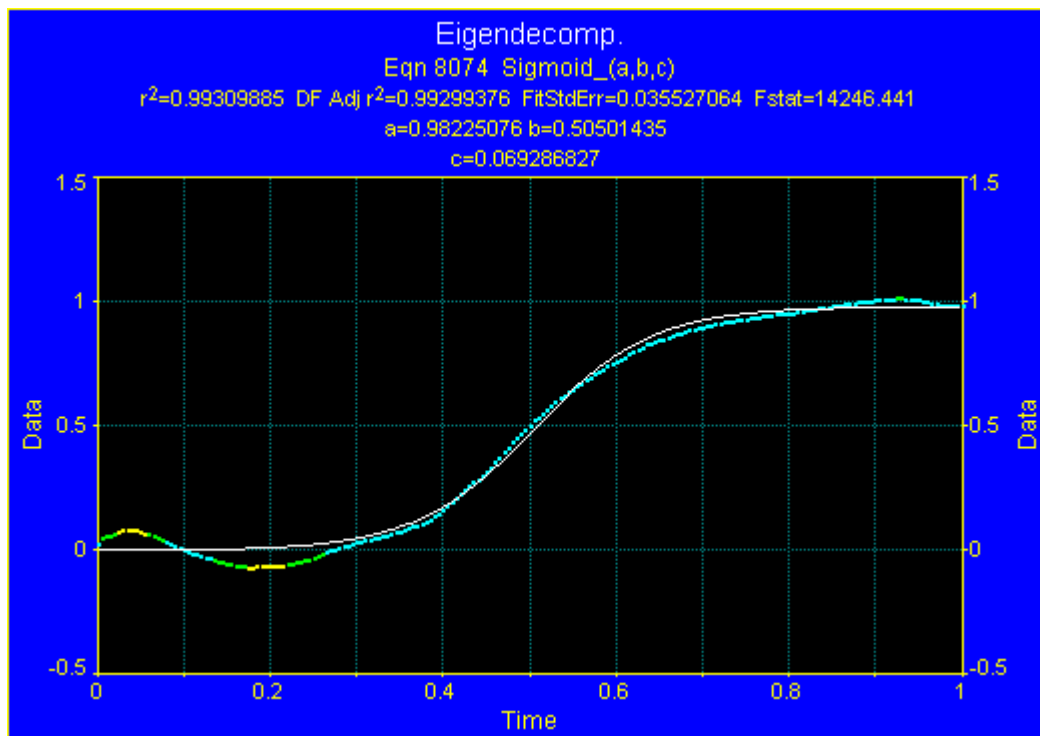
Select the **Kaiser-Bessel** algorithm and click on the **AI Expert** button.

This is a very difficult data set to smooth because of the high noise level and a transition function's unique

mix of straight line and curved regions. We will list the results for all six, although for this tutorial we will proceed with the one algorithm that best preserves the transition shape visually.

Select the **Eigendecomp.** algorithm. Click the **OK** button. Answer **Yes** to update the data table.

Select the **Process** menu's **Curve Fit Transition Functions** and click the **Fit** button. When the fit is concluded, enter the **Review** by clicking the **Graph Start** button.



The coefficients of 0.982, 0.505, and 0.069 represent errors of 1.8, 1.0, and 7.6% from the noise-free values.

Comparing the Algorithms

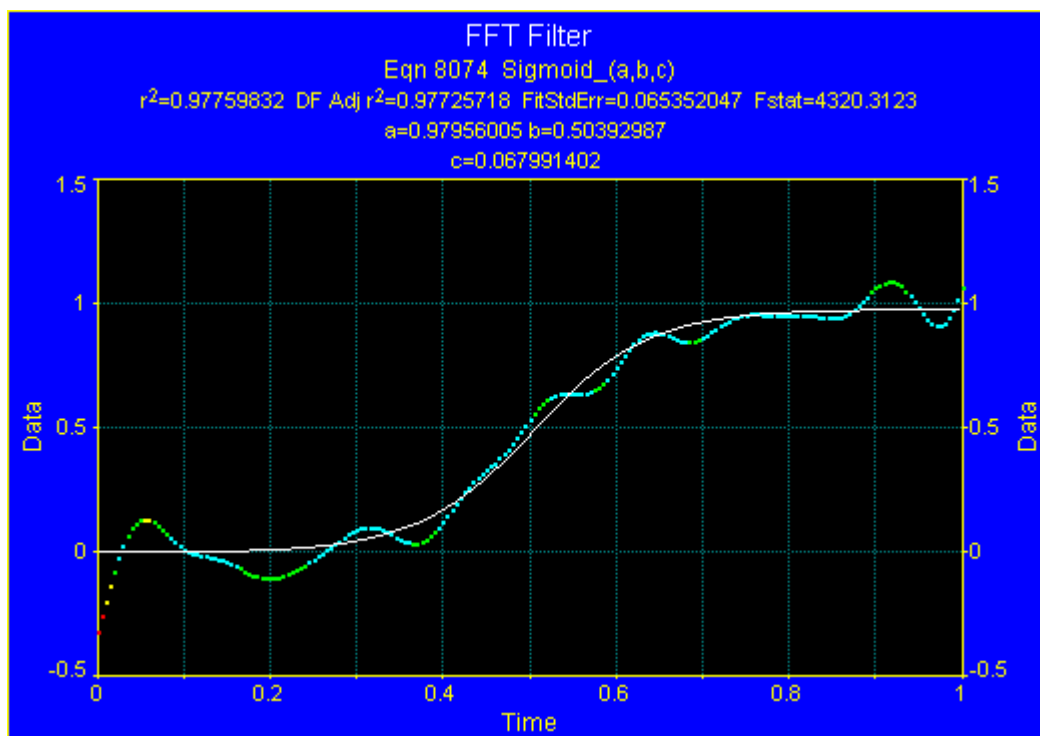
To create a simple appraisal, we will average together the errors for the three coefficients.

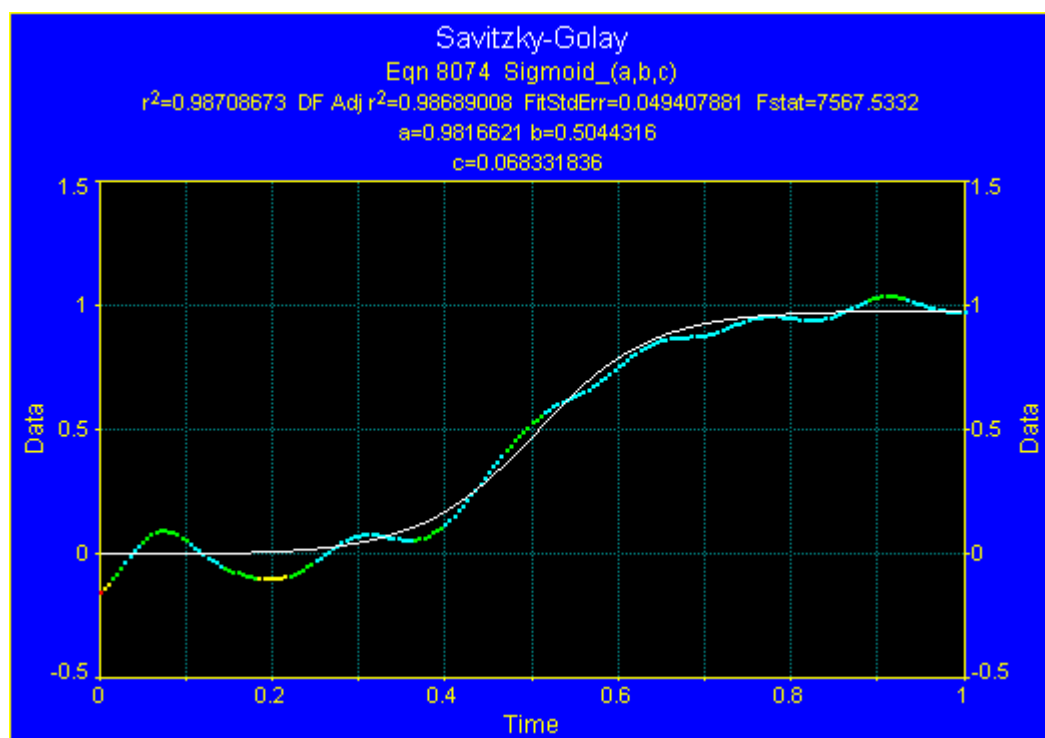
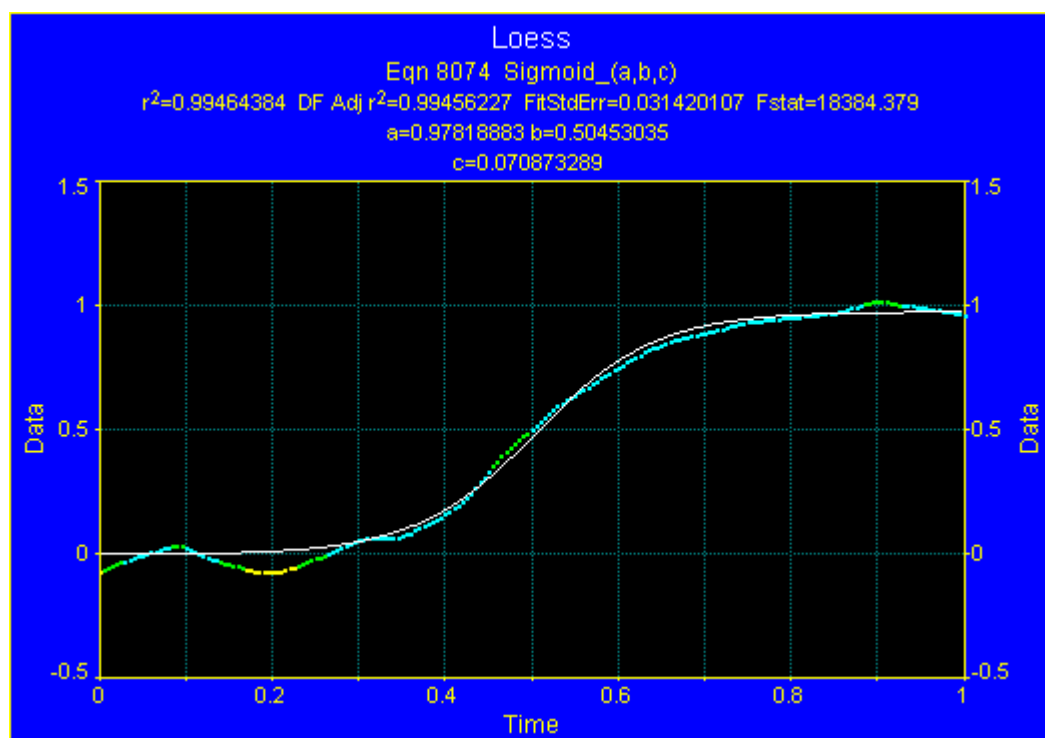
Parametric, all points (0.9783, 0.5037, 0.0678)	ParmErr=4.17%	$r^2=0.7965$	EqNoise=57.8%
Parametric, excl 2SD+ (0.9869, 0.5088, 0.0719)	ParmErr=2.40%	$r^2=0.8273$	
FFT Filtering (0.9796, 0.5039, 0.0680)	ParmErr=4.05%	$r^2=0.9776$	
EqNoise=0.182%			
Loess (0.9782, 0.5045, 0.0709)	ParmErr=2.85%	$r^2=0.9946$	
EqNoise=0.195%			
Savitzky-Golay (0.9817, 0.5044, 0.0683)	ParmErr=3.88%	$r^2=0.9871$	
EqNoise=0.181%			
Gaussian Deconv. (0.9546, 0.4950, 0.0635)	ParmErr=6.96%	$r^2=0.9759$	
EqNoise=0.179%			
Eigendecomp. (0.9822, 0.5050, 0.0693)	ParmErr=3.46%	$r^2=0.9931$	
EqNoise=0.294%			
Kaiser-Bessel (0.9867, 0.5058, 0.0700)	ParmErr=3.05%	$r^2=0.9915$	
EqNoise=0.179%			

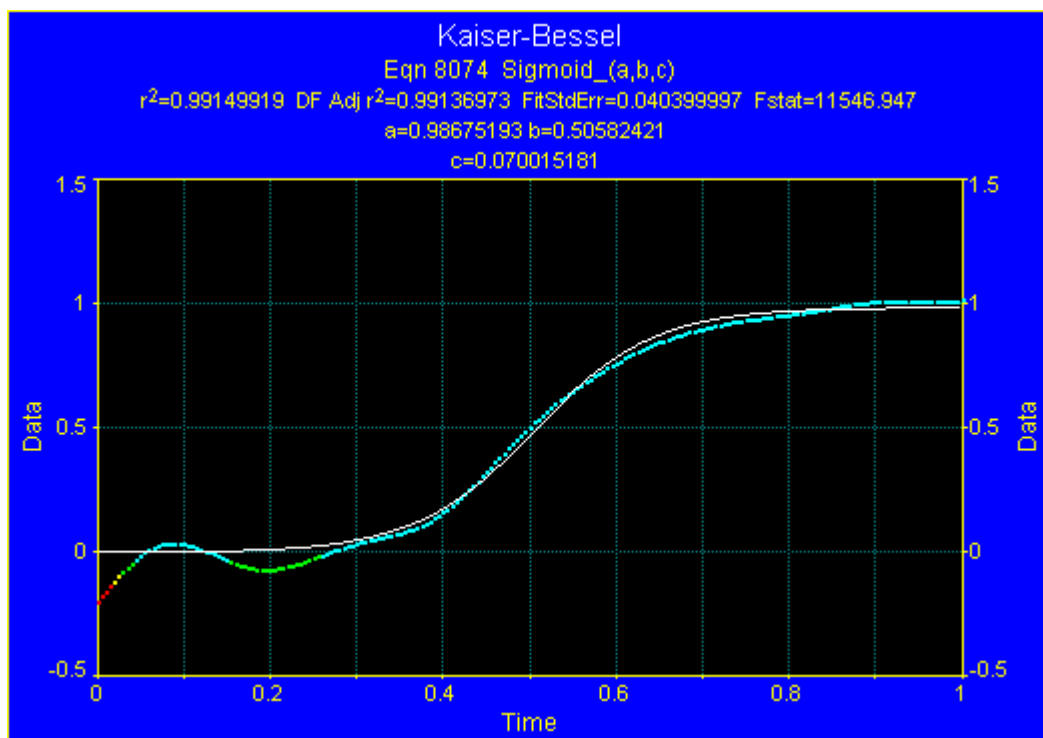
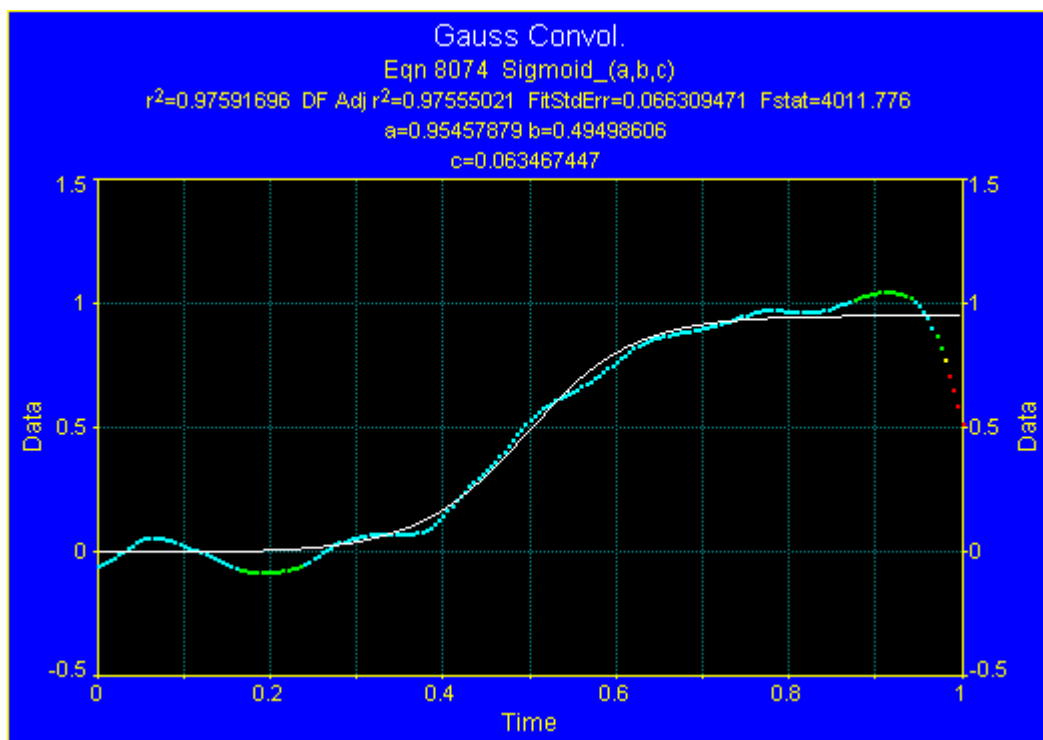
Apart from the small attenuation observed with the Gaussian Deconvolution smoothing, the parametric fit of the smoothed data produces parameter values very close to that of the least-squares sigmoidal fit of the unsmoothed data. The high r^2 , particularly with the Loess and Eigendecomposition algorithms, indicates a very high level of smoothing was achieved while minimally impacting the underlying data trend. The smoothing algorithms included within TableCurve 2D have been selected for their robustness and effectiveness. They represent the best in noise removal technology.

The Equivalent Noise % uses a time domain polynomial interpolation procedure for estimating the Gaussian noise in a data set. The 57.8% white noise estimate for the raw data compares favorably with the 50% noise actually added. All of the procedures filtered out at least 99.5% of the noise originally present.

Here are the curve-fits from the other five procedures:







The AI Expert in the [Automated Smoothing](#) seeks to select reasonable smoothing levels. It cannot select an optimum. Visually, there is at least a promise that the FFT and Savitzky-Golay procedures could be further optimized.



Click OK to close the Review.



Click the **Reset XY Data** to restore the unsmoothed data.

Savitzky-Golay Smoothing



Select the **Savitzky-Golay Smoothing** procedure from the main toolbar or Filter menu.

Set the one-sided size of the moving window (**Win n**) to **32**. Set the **Order** to **2** for quadratic fitting and set the sequential pass count (**Passes**) to **3**. Be sure **Data** is selected.

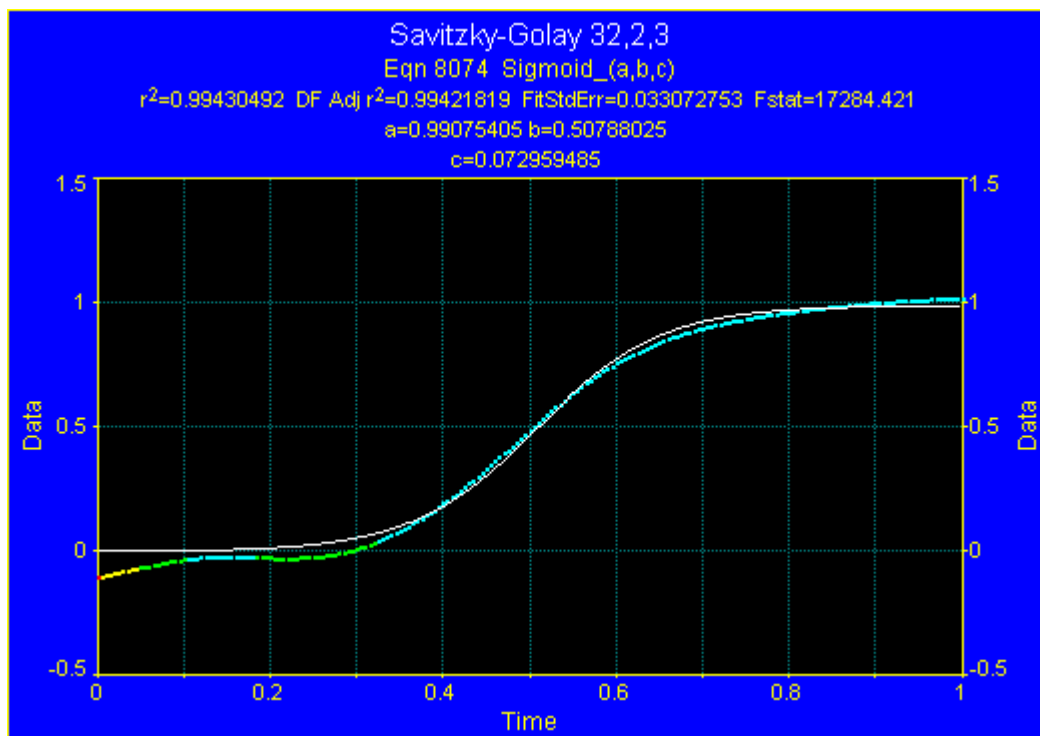
Note that the quadratic does a better job of managing the straight line portions of the transition.



Click **OK** to close the procedure. Answer **Yes** to update the data table.



Again select the **Process** menu's **Curve Fit Transition Functions** and click the **Fit** button. When the fit is concluded, enter the **Review** by clicking the **Graph Start** button.



Savitzky-Golay (0.9817,0.5044,0.0683)	ParmErr=3.88%	$r^2=0.9871$
EqNoise=0.181%		
Savitzky-Golay Opt. (0.9907,0.5078,0.0729)	ParmErr=1.74%	$r^2=0.9943$
EqNoise=0.177%		

The optimized algorithm offers a significant improvement in noise reduction and better recovers the underlying parameters.



Click **OK** to close the **Review**.



Click the **Reset XY Data** to restore the unsmoothed data.



Select the **Fourier Denoising** option in the Filter menu or main toolbar.

Experiment with different windows (data tapers). Explore both **Frequency** and **dB thresholds**. Graphically set the thresholds by clicking at the frequency or signal magnitude desired.

Fourier denoising depends on the ability to attribute all spectral information beyond a certain frequency or below a certain spectral threshold to noise. Because of spectral leakage from edge effects and components failing to center within a frequency bin, the signal is not likely to be conveniently confined to certain bins while noise is confined exclusively to the remainder. In practice, the signal/noise separation is incomplete.

Finding a good signal-noise threshold, either in frequency or signal strength, is not easy with this data set. You will probably be disappointed. Sums of sinusoids do not easily map a mix of curves and straight line regions within a data stream.



Click **Cancel** to close the procedure.

Eigendecomposition Denoising

An eigendecomposition uses adaptive non-parametric basis functions, partitioning a data stream by signal strength rather than frequency. The highest power components distribute amongst the initial eigenmodes while the noise is found in the subsequent eigenmodes. Denoising using eigendecomposition procedures is often the best way to remove noise from data, provided a high enough matrix decomposition order is possible.

A single eigenmode will capture a continuously increasing or continuously decreasing component within a data stream. An exponential decay, for example, is captured by a single eigenmode. Two eigenmodes are needed to capture a transition since the slope is not continuously increasing or decreasing. Two eigenmodes are also needed to capture an oscillation of a given frequency. That oscillation can be a sinusoid, an exponentially damped sinusoid, a sawtooth, a square wave, or any other shape. If the amplitude of the oscillation varies in accord with a continuous function of time, the two eigenmodes will still map this oscillation. On the other hand, variable frequency oscillations such as chirps may require a large number of eigenmodes to fully map the non-stationary data trend.

Select the **Eigendecomposition Denoising** option from the Filter menu or main toolbar. Be sure the **CovM FB (forward-backward covariance matrix)** algorithm is selected and set the **Order of the eigendecomposition matrix to 70**. To accommodate the single transition, set the **Signal Space to 2**.

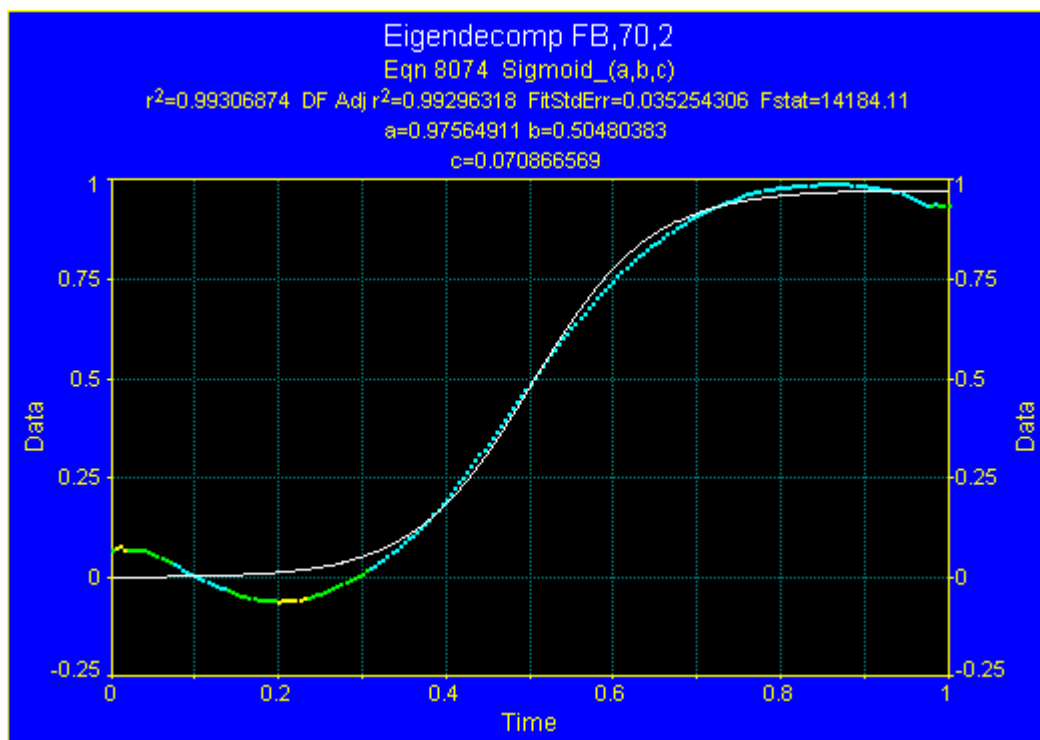
Note that you can set the signal space graphically by selecting the last eigenmode representing signal. Do not include eigenmodes that are part of the long sloping noise floor.



Click **OK** to close the procedure. Answer **Yes** to update the data table.



Once again, select the **Curve Fit Transition Functions** item and click the **Fit** button. When the fit is concluded, enter the Review by clicking the **Graph Start** button.



Eigendecomp. (0.9822, 0.5050, 0.0693)	ParmErr=3.46%	$r^2=0.9931$
EqNoise=0.294%		
Eigendecomp. Opt (0.9756, 0.5048, 0.0709)	ParmErr=2.96%	$r^2=0.9931$
EqNoise=0.236%		

The higher order eigendecomposition allowed for more of the noise to be captured and discarded. In the automated smoothing, the eigendecomposition's order is never greater than 40. In this instance, a model order of 70 means that 68 eigenmodes are available to capture and discard noise, rather than 38. This higher level of denoising resulted in a better recovery of the underlying parameters, although the goodness of fit was not improved because of the larger oscillation on the lower plateau of the transition.



Click on **OK** to exit the Review. Exit the program by closing the main window or by the **Exit** item in the File menu.

Derivative Estimation

In this tutorial, we will explore the different procedures TableCurve 2D offers for generating accurate estimates for derivatives.

Approaches

There are five ways to estimate derivatives in TableCurve 2D:

- Fit a parametric model and generate analytic or numeric first and/or second derivative estimates. For a parametric model, noise is not a factor. The derivatives are computed from the curve-fit equation. If the default analytic derivatives are set, the generated derivatives will contain full numeric precision. Accuracy, however is a different matter. The derivatives will only be as accurate as the model selected to fit the data.
- Fit a non-parametric smoothing spline to the data and evaluate the analytic derivatives of the spline. A spline's derivatives are strongly influenced by noise. If any noise at all is present within a data stream, some form of smoothing will be needed. As with least-squares parametric fitting, the spline coefficients for a smoothing spline are computed in an overspecified sense. The number of knots (nodes where segments meet) in the spline will be fewer than the number of data values.
- Fit an interpolating spline to data that have been pre-smoothed and compute the derivative analytically from the spline. In this instance, the number of nodes in the spline will be equal to the data count. For this approach to work effectively, a very high level of smoothing is required.
- Perform a Fourier decomposition of data and compute derivatives from the individual sine components. Fourier interpolation is slow and highly sensitive to noise. A Fourier procedure must be combined with a lowpass filter in the frequency domain in order to produce reasonable derivative estimates.
- Locally fit polynomials with a sliding window across the data and directly create a derivative filter of the desired order. A Savitzky-Golay derivative filter is usually the most accurate non-parametric approach for estimating derivatives. It is also the only recommended approach for derivatives beyond second order.

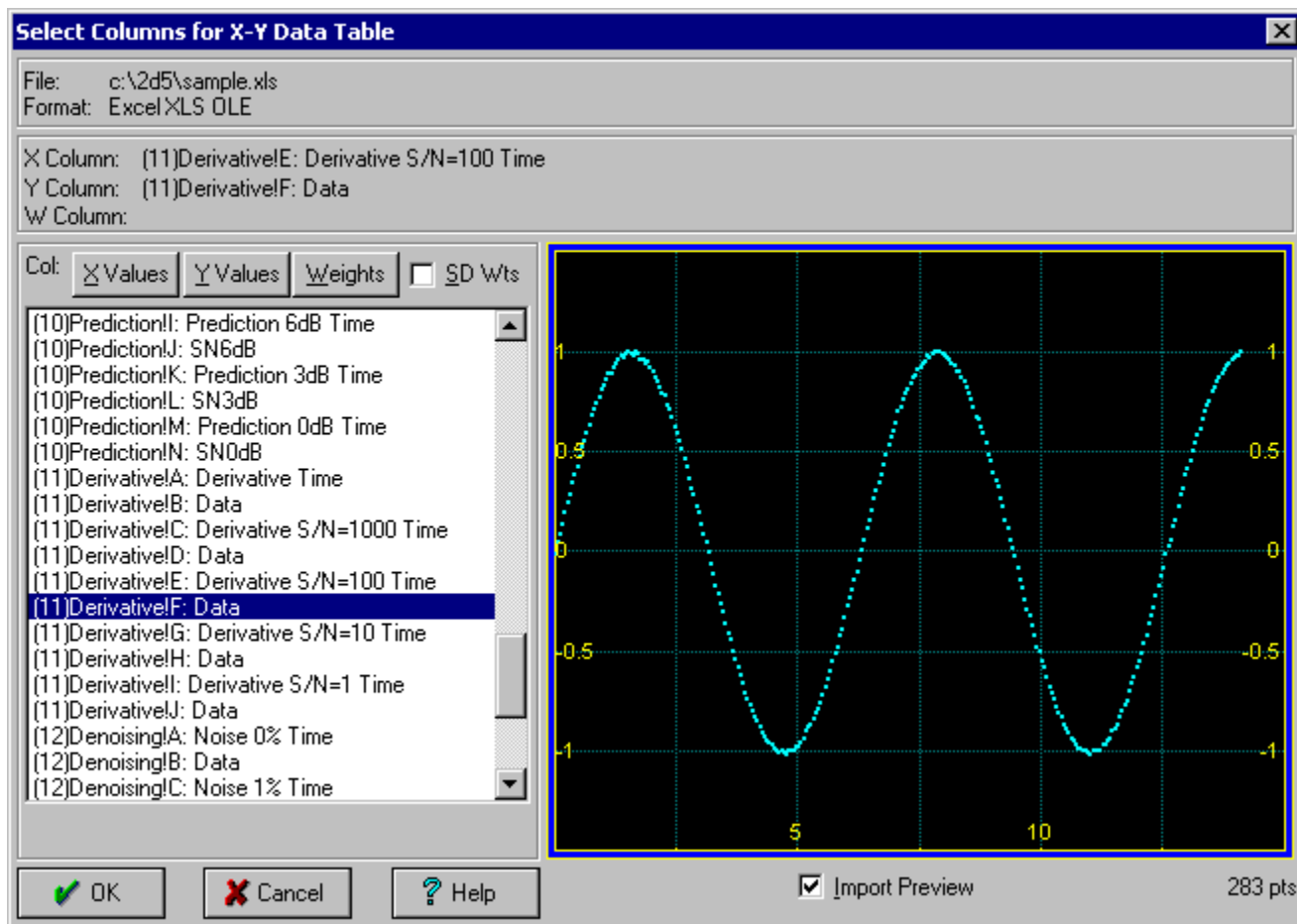
Importing A Noisy Sinusoid Data Set

Start TableCurve 2D. Select **Start/Programs/TableCurve 2D v5**.



Select the File menu's **Import** option (or use the **Import** button in the main toolbar). If Excel [xls] files are not shown, click on the **Files of Type** drop-down button and select **Excel [xls] files**. Select and open the file **SAMPLE.XLS**.

Select the third column with the label **(11)Derivative!E: Derivative S/N=100 Time to be used as the X-variable in the data table**. Select the fourth column identified in the selection list as **(11)Derivative!F: Data for the Y-variable**.



Press **OK** to accept these choices. Press **OK** once again within the titles dialog to confirm the imported titles.

This sinusoid contains 1% added Gaussian noise. This may sound very small, but it is enough to render most derivative estimation algorithms useless, especially for derivatives beyond first order. The data are scaled so that the second derivative should be a cosine wave rather than a sine wave, and equal in both magnitude and phase. Also, the fourth derivative should exactly overlay the original sine wave. This is the classic test case for evaluating the accuracy of derivative algorithms.

Fitting a Parametric Model



Click the **Curve Fit Waveform Functions** option in the **Process** menu or toolbar.

When the fit is concluded, enter the **Review** by clicking the **Graph Start** button.



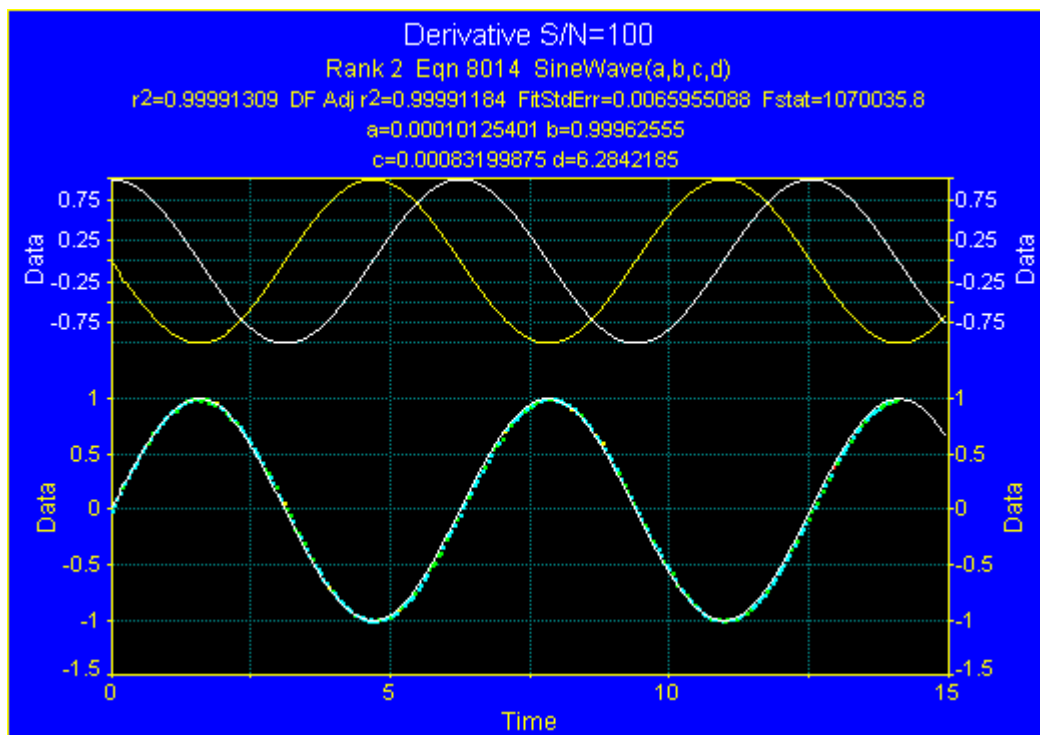
As an alternative, you may press **OK** and select the **Graph Start** item in the **Review** menu. If you do not respond to the conclusion of the fit in 10 seconds, the **Review** is automatically started.

In the **Equation List**, select **Equation 8014**, the sine wave model.



Click the **Set Y2 Axis Content** button and select **Curve-Fit, Data in Y1; First, Second Derivatives in Y2**. Click **OK**.

Select the **Preferences** option in the **File** menu of the **Curve-Fit** graph. Be sure the **Analytic Curve-Fit Equation Derivatives** is checked. Click **OK**.



This sine wave data set was generated with a zero mean, an amplitude of 1.0, a phase of 0, and a wavelength of 2π . Note that the parameters are recovered to about four decimal places. The white curve in the Y2 plot is the first derivative and the yellow curve is the second derivative. Note that the second derivative is identical to the data except for the sign.

Derivatives are generated:



In the **Excel**, **Lotus**, and **SigmaPlot** export options



In the **Copy** procedure of the Curve-Fit graph



As one of the **Generate Table** options in the **Evaluation** procedure

To test the accuracy of the second derivative, we will copy the curve-fit information to the clipboard. Second derivative evaluations are automatically included.



Click the **Copy** button in the control panel of the Curve-Fit graph.



Click the **Set Y2 Axis Content** button and select **Curve-Fit,Data** in **Y1**. Click **OK**.



Click **OK** to close the **Review**.

To make this data more convenient for repeated import, we will save it as an SPSS file. This data can then be reimported in one step by selecting the file from most recently used list in the **File** menu.



Click the **Save As** button in the main toolbar. Set the **Save As Type** to **SPSS [sav]** and enter **tour4**. Click **OK**.



Select the **Import Clipboard** button or **File** menu item. Select **Column K: X Observed** for the **X** variable and **Column U: Second Deriv** for the **Y** variable. Click **OK** to initiate the import. Click **OK** to accept the titles.

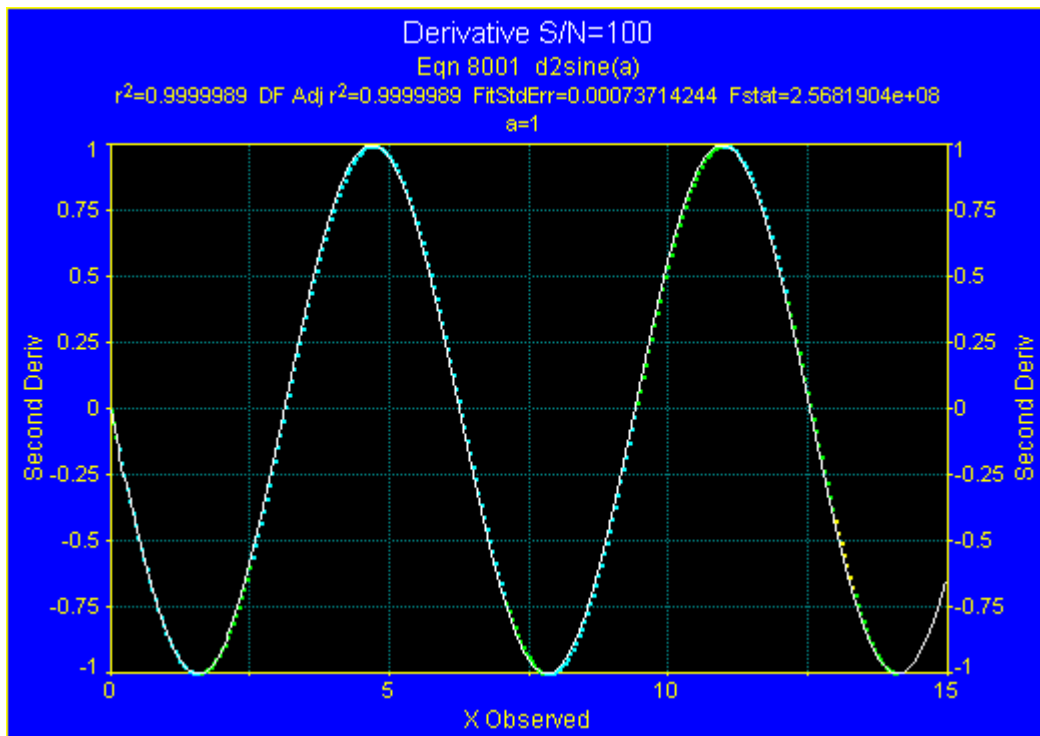
We will now employ a little trick to fit this second derivative data to the expected underlying parametric function, the second derivative of the sine function with an amplitude of 1.0, a phase of 0.0, and a wavelength of 2π .



Select the **User Functions** option in the **Process** menu or toolbar. Enter **d2sine** for the name, enter **1** for the parameter count, enter **$Y=A0*0-SIN(X)$** as the function expression. Click **OK** to exit the UDF procedure and reply **OK** to the message that indicates **A0** was found to be insignificant.



Select the **Curve-Fit User Functions** option in the **Process** menu or toolbar. When the fit is concluded, enter the **Review** by clicking the **Graph Start** button.



Since none of the parameters are adjusted in the non-linear fit, we compute a correlation with the underlying second derivative function. The standard error is 0.00074. Thus a non-linear least-squares sine fit generates an analytical second derivative accurate to this measure.

We will now generate second derivative data sets using each of the other four approaches for estimating derivatives. We will then similarly assess the accuracy of these estimations.



Click **OK** to close the **Review**.

Non-Parametric Smoothing Spline

In this approach, we will fit a smoothing spline to the data and evaluate the analytic derivatives of the spline.

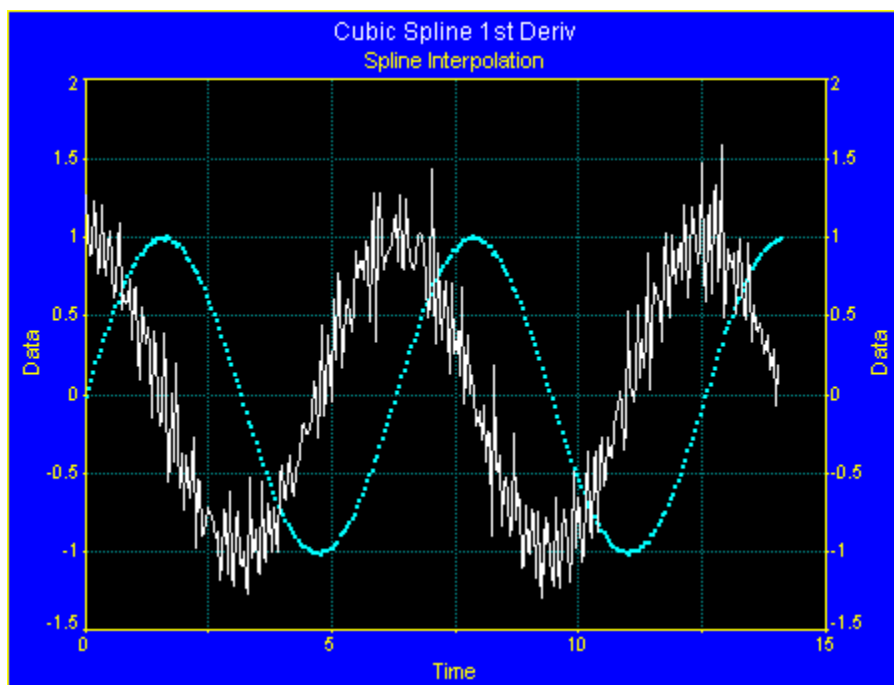
Select **tour4.sav** from the first position of the most recently used files list in the File menu. Answer **No** to save the existing data and click **OK** to accept the titles.



Select the **Spline Estimation** option in the Process toolbar or Estimate menu.

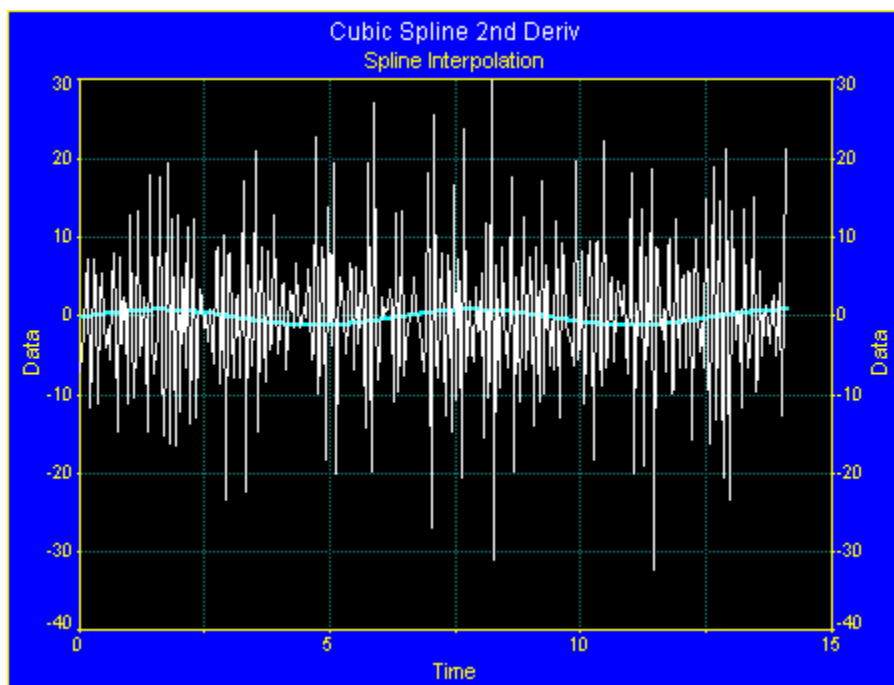
Select the **Cubic** procedure, set Output Data to **1st Deriv**, and set **n** to **283** to exactly match the incoming data.

By default, the **n** is set to perform a 2x upsampling.



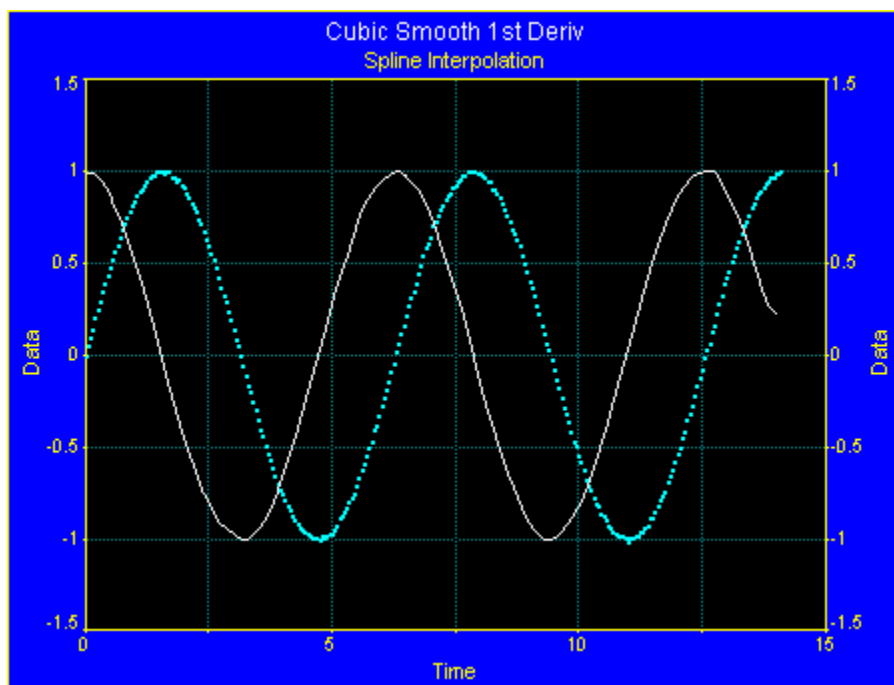
Even at a 1% noise level, the first derivative of an interpolating spline is barely stable.

Set Output Data to **2nd Deriv**.



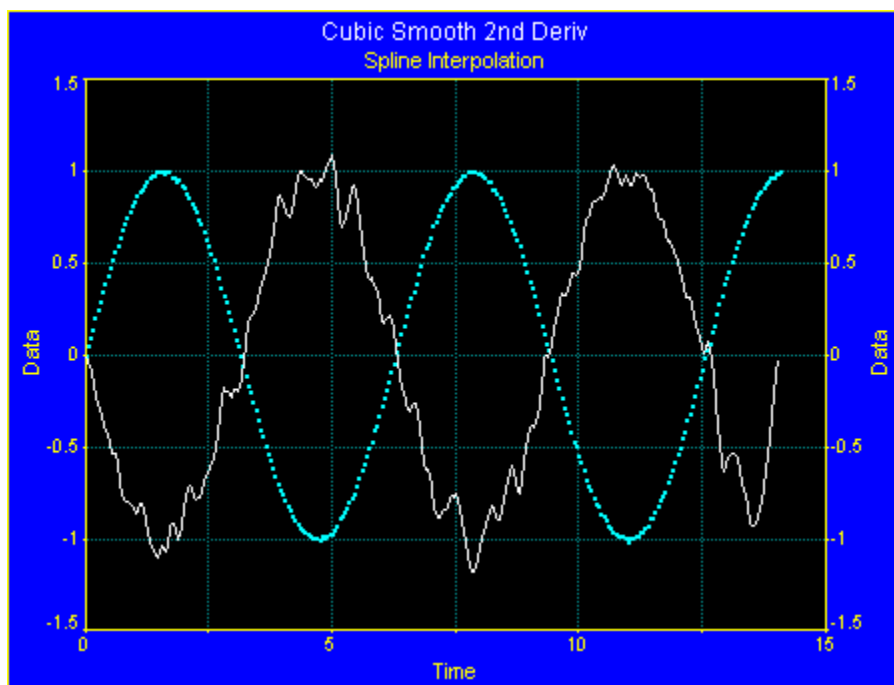
All accuracy is lost.

Select the **Cubic smooth** procedure. Set Output Data to **1st Deriv**.



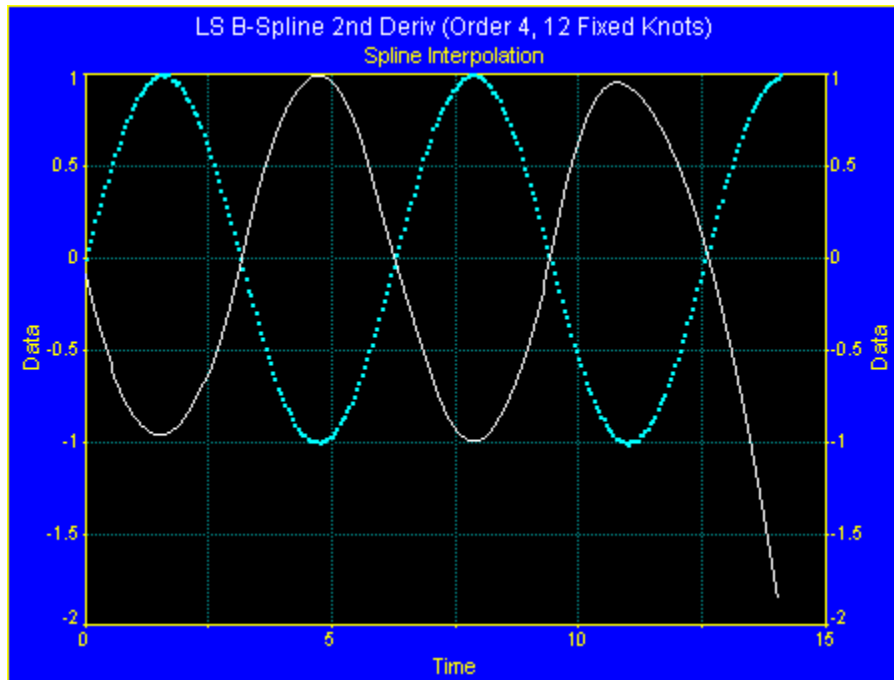
This "cross-validation" smoothing spline is probably the most widely used of all smoothing splines. A cubic (order 3) spline produces smooth first derivatives and continuous (but not smooth) second derivatives.

Set Output Data to **2nd Deriv**.



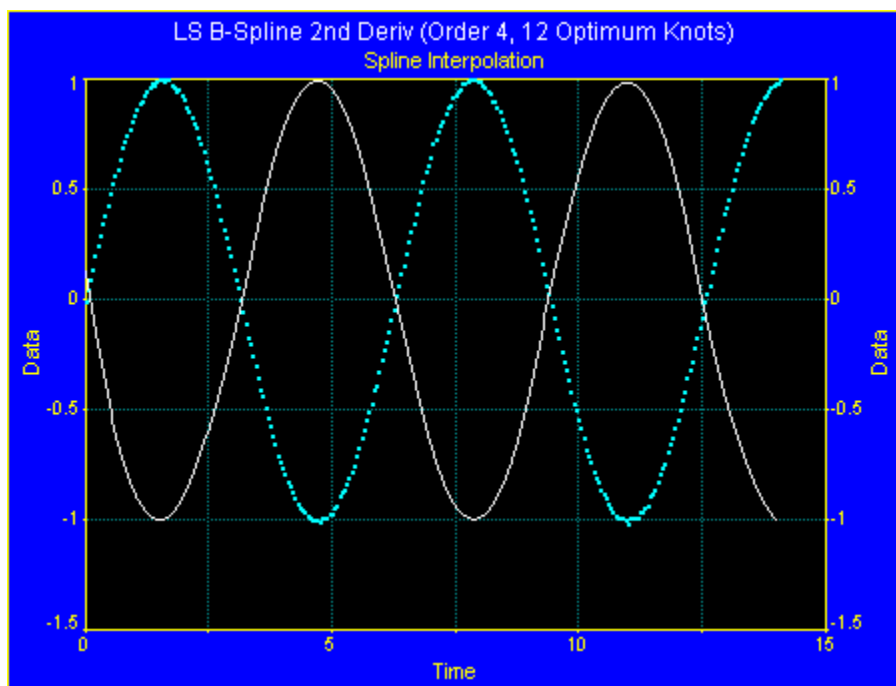
Since most users want both continuity and smoothness, the **Cubic Smooth** procedure is generally only of value for first derivatives. To get smooth second derivatives, a quartic (order 4) spline is needed.

Select the LS B-Spline fix procedure. Set the Order to 4 and the Knots to 12.



The shape is irregular because the knots are not placed at the crests, troughs, and inflection points. There are two options, a non-linear optimization that seeks to find these optimum knots, and a direct specification of them.

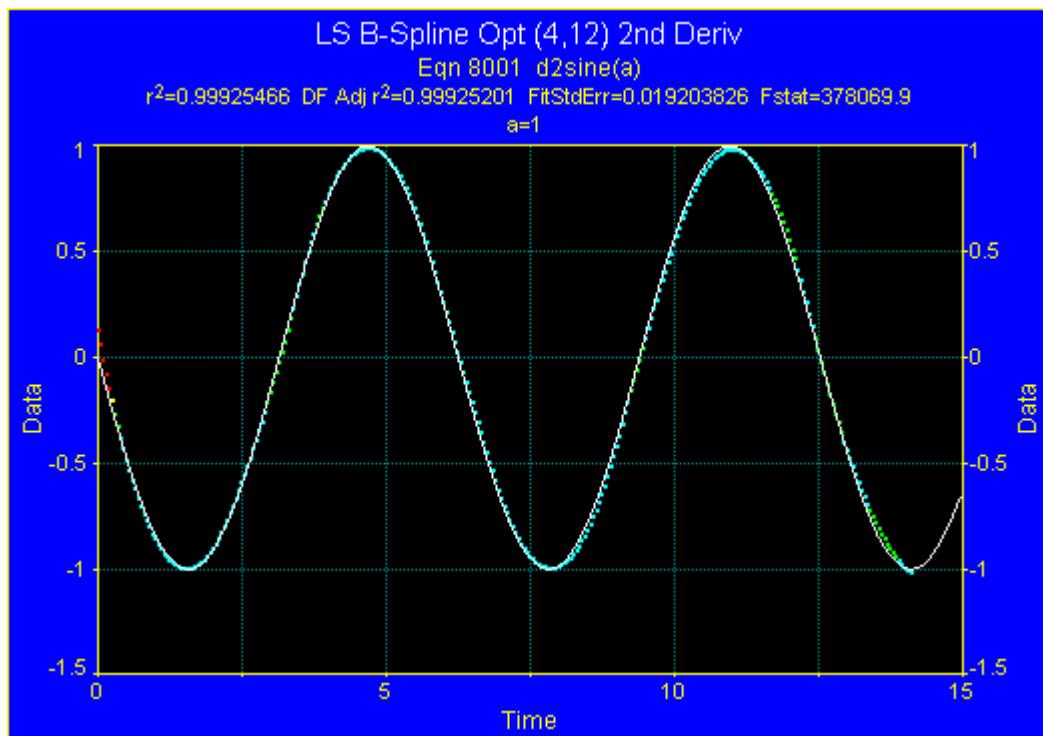
Select the **LS B-Spline opt** procedure.



Click OK to close the estimation procedure and answer **Yes** to updating the data table.



Select the **Curve-Fit User Functions** option in the **Process** menu or toolbar. When the fit is concluded, enter the **Review** by clicking the **Graph Start** button.



The second derivative's standard error relative to the underlying generating function is 0.0192.



Click **OK** to close the Review.



Click **Reset XY Data** to restore the original sinusoidal data with the 1% noise.

Fitting an Interpolating Spline to Pre-smoothed Data

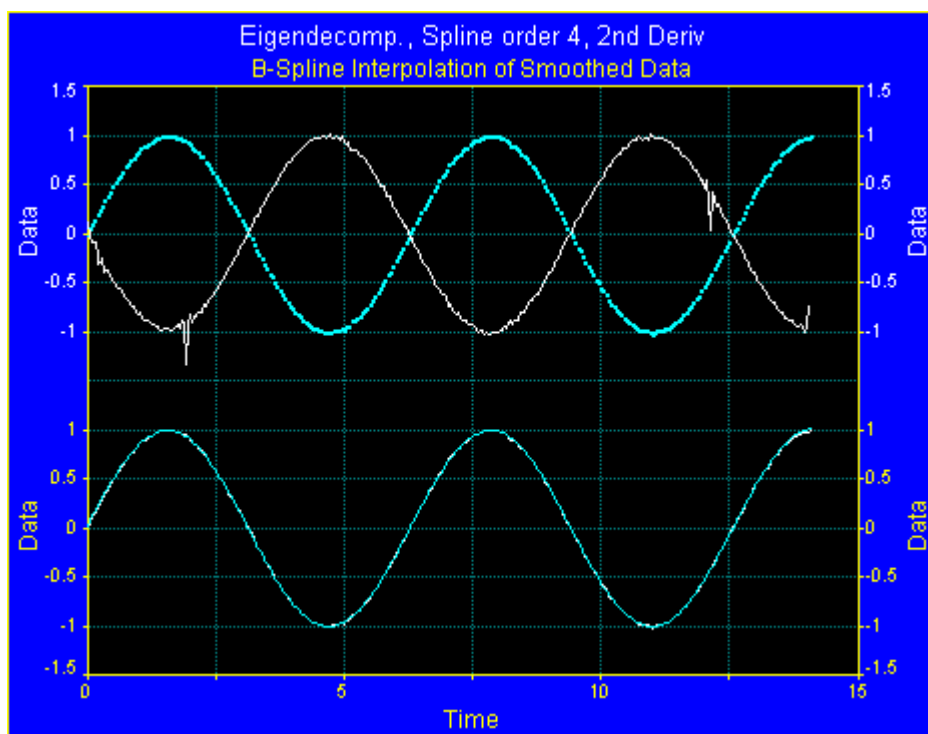


Select the **Smoothed Data Spline Estimation** option in the Estimate menu or Process toolbar.

Select the **Eigendecomp.** procedure, set the B-spline order to **4**, set output data to **2nd Deriv**, and set **n** to **283**.



Click the **AI Expert** button.



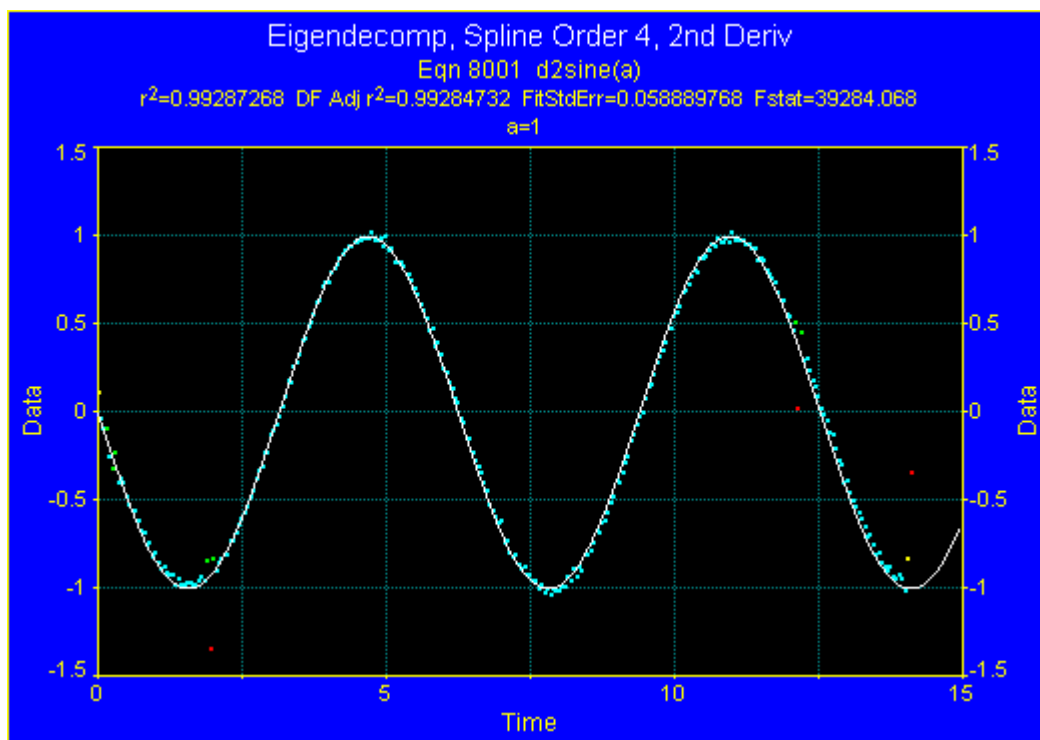
In all cases, including Savitzky-Golay, this procedure smooths the raw data, a B-spline is fitted to the smoothed data, and analytic derivatives are computed from the B-spline interpolant. Note that the zero order smoothing in the lower plot looks good for most of the algorithms. The analytic second derivatives in the upper plot are not going to win many awards, although the Eigendecomp. procedure is probably the most accurate.



Click **OK** to close the estimation procedure and answer **Yes** to updating the data table.



Select the **Curve-Fit User Functions** option in the Process menu or toolbar. When the fit is concluded, enter the Review by clicking the **Graph Start** button.



The second derivative's standard error relative to the underlying generating function is 0.0588.



Click OK to close the Review.



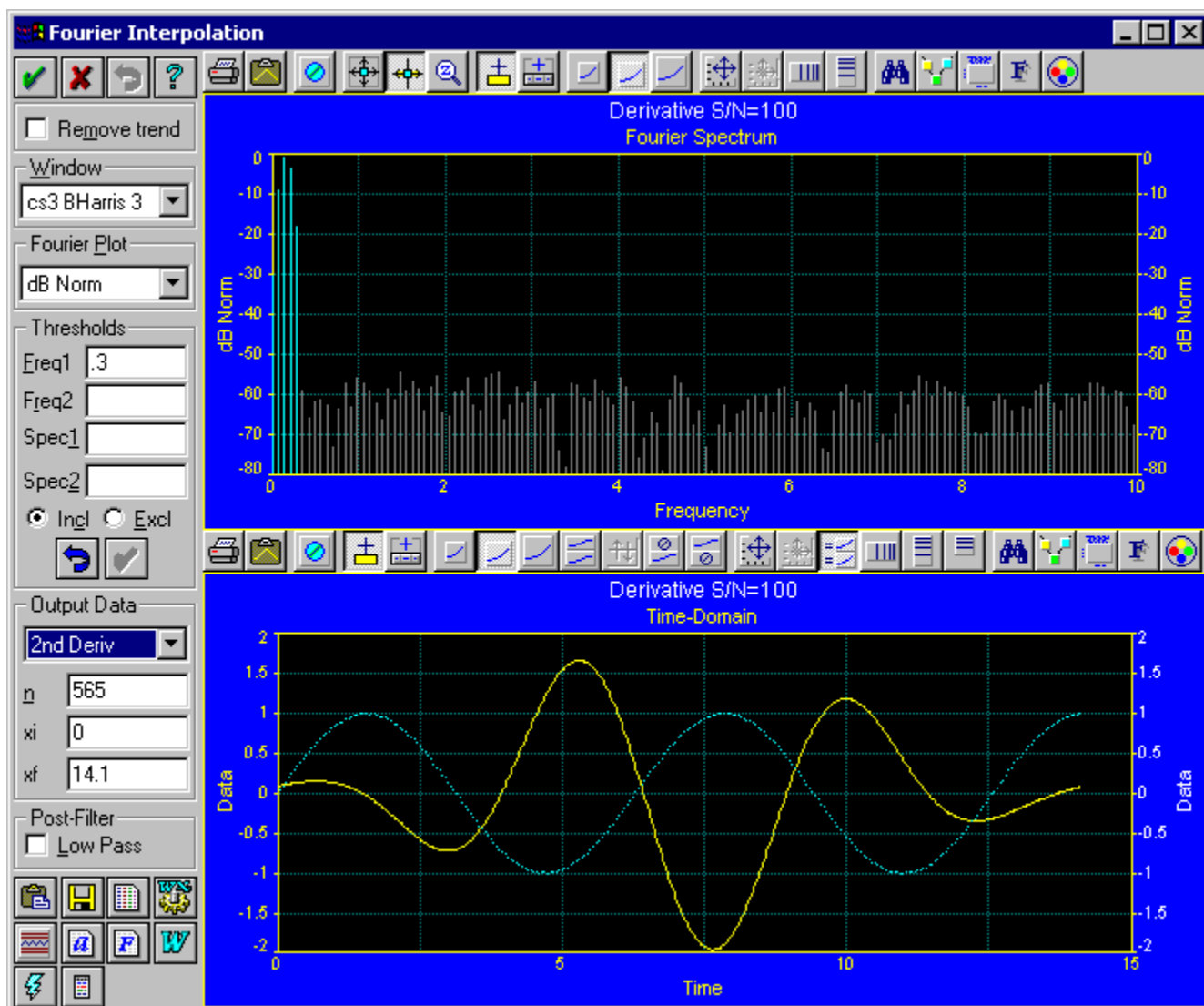
Click Reset XY Data to restore the original sinusoidal data with the 1% noise.

Fourier Derivative Estimation



Select the Fourier Estimation option in the Estimate menu or Process toolbar.

Select the cs3 BHarris 3 window, enter 0.3 for Freq1. and set the Output Data to 2nd Deriv.



Note that a data taper is needed to isolate the harmonic and filter the noise. The second derivative is smooth and hints at the true shape, but accuracy is very poor.

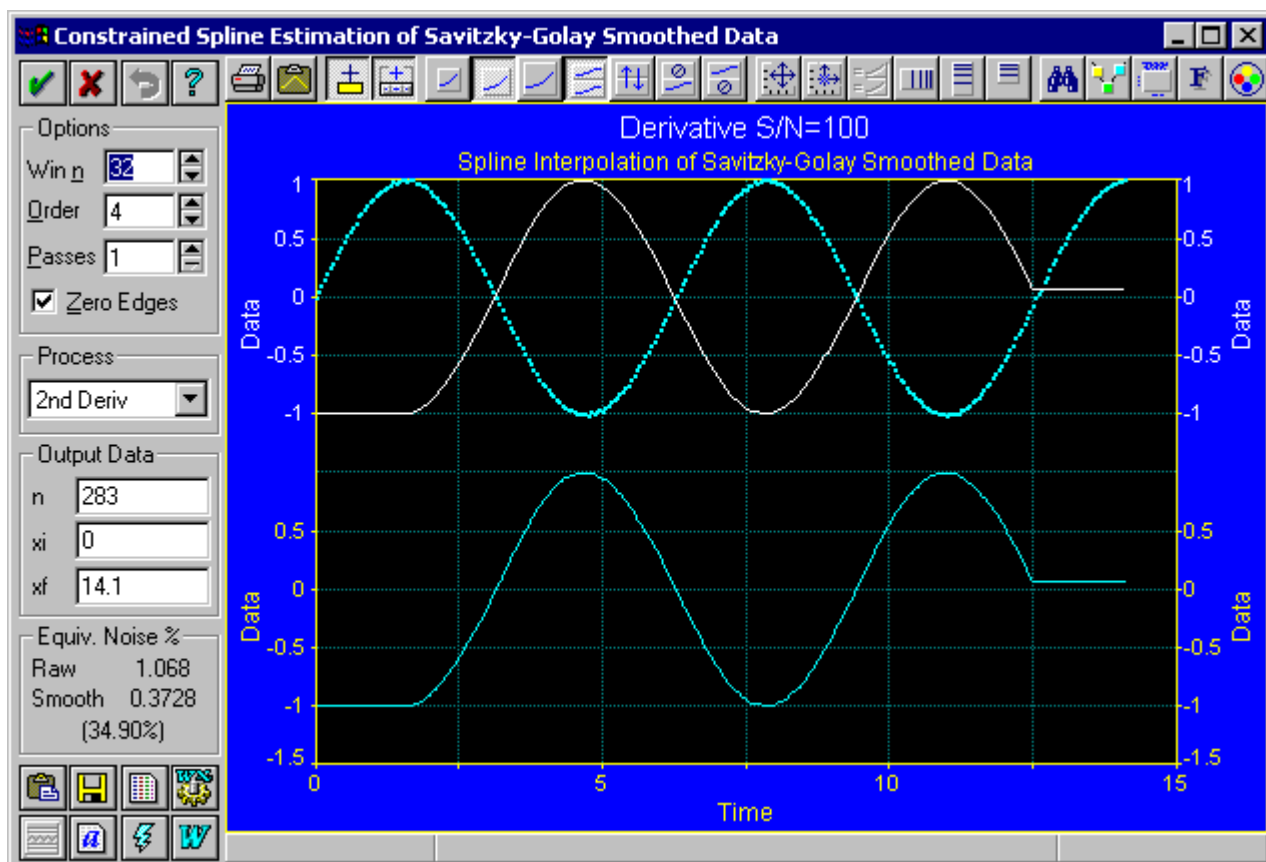
 Click **Cancel** to exit the Fourier Estimation procedure.

Savitzky-Golay Derivative Filters

 Select the **Savitzky-Golay Spline Estimation** option in the **Estimate** menu or **Process** toolbar.

This option creates a filter that smooths directly to the derivative order of choice. A constrained cubic spline is then fit to the derivative data in order to provide continuous estimates. This procedure was designed for accuracy in derivatives.

Set the one-sided sliding window (Win n) to **32**, be sure the order is **4**, and set Passes to **1**. Check the **Zero Edges** box. Set Process to **2nd Deriv** and set n to **283**, the size of the incoming data.



The Zero Edges option treats edge effects in the filter by holding the estimate constant at the last value where edge effects are absent. In general, a properly designed Savitzky-Golay derivative filter will be very accurate except in these edge effect regions. We will now change the exported range so that these edge effect regions are excluded.

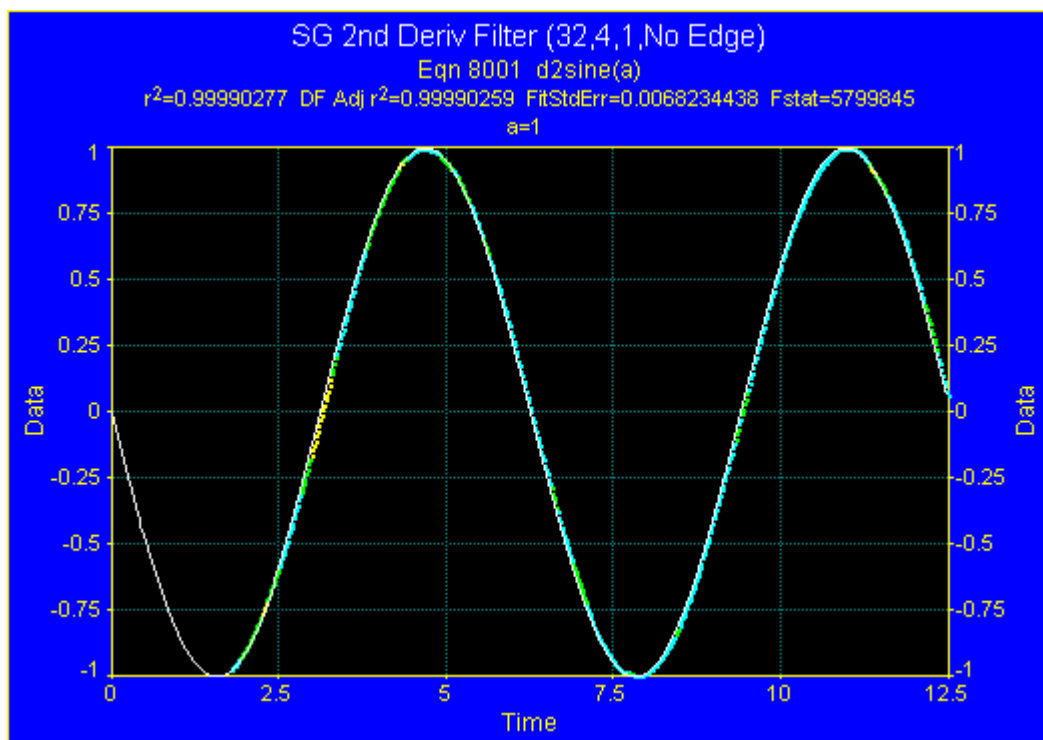
Set xi to 1.8 and xf to 12.5.



Click **OK** to close the estimation procedure and answer **Yes** to updating the data table.



Select the **Curve-Fit User Functions** option in the **Process** menu or toolbar. When the fit is concluded, enter the **Review** by clicking the **Graph Start** button.



The second derivative's standard error relative to the underlying generating function is 0.0068. This is an excellent accuracy.



Click OK to close the Review.

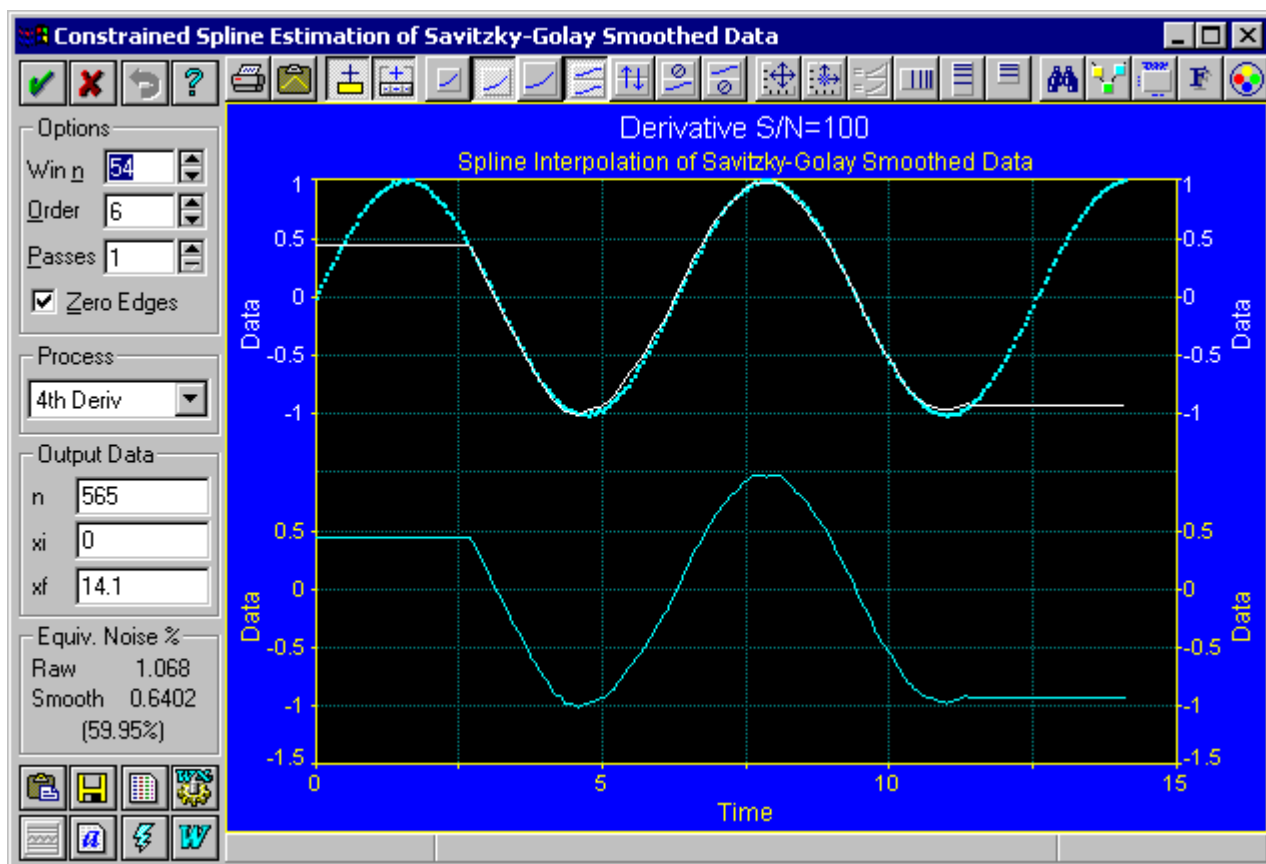


Click Reset XY Data to restore the original sinusoidal data with the 1% noise.



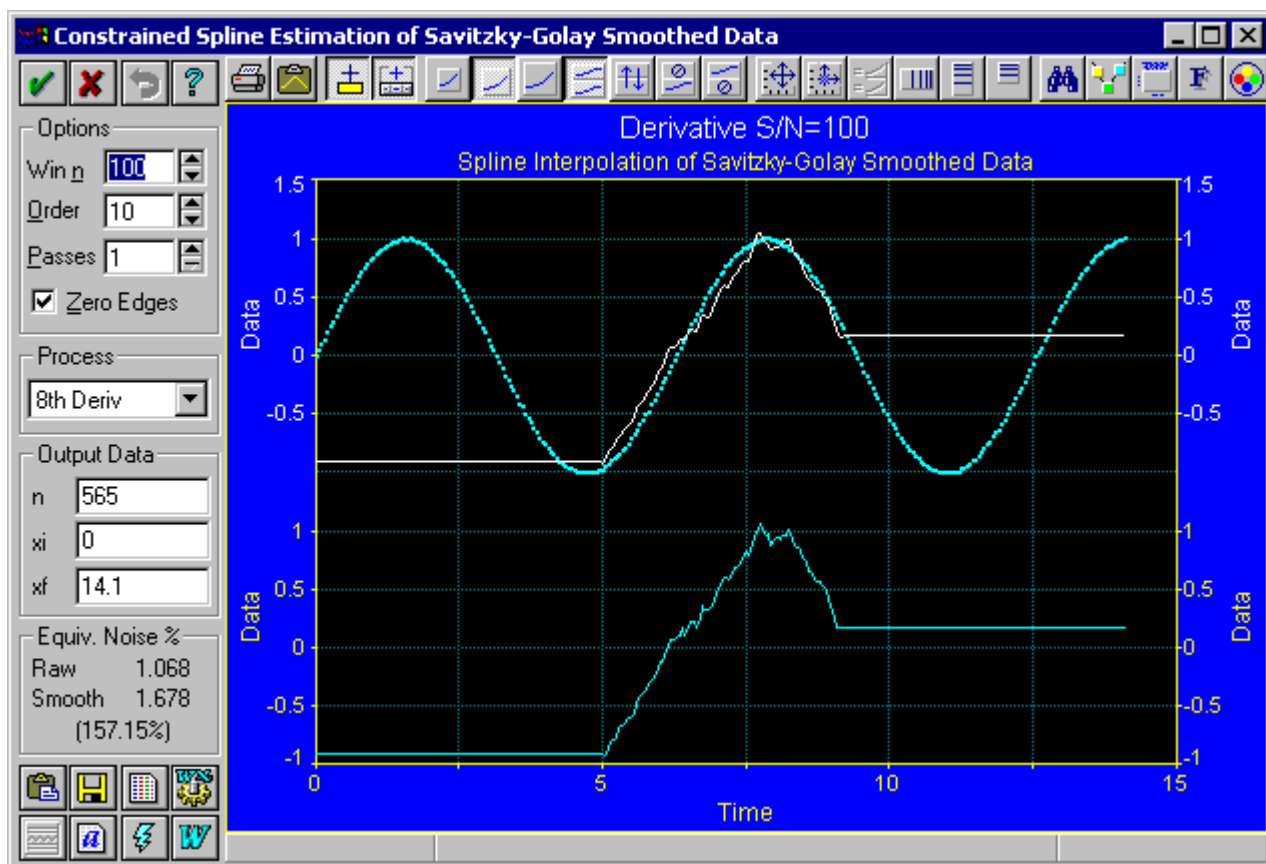
Again select the Savitzky-Golay Spline Estimation option in the Estimate menu or Process toolbar.

Set the one-sided sliding window (Win n) to 54, set the order is 6, and be sure Passes is set to 1 and the Zero Edges box is checked. Set Process to 4th Deriv.



Although you will generally need to remove the edge effect regions and go to higher window sizes, it is possible with this procedure to get accurate higher order derivatives. The fourth derivative of a scaled sine wave is the original function. Very few derivative algorithms are capable of accurately resolving a fourth derivative.

Set the one-sided sliding window (Win n) to 100, set the order is 10, and be sure Passes is set to 1 and the Zero Edges box is checked. Set Process to 8th Deriv.



Like the fourth derivative, the eighth derivative of a scaled sine wave will be the original waveform. Given the size of the data set, the zone free of filter edge effects is small. Still, the eighth derivative is valid and has reasonable accuracy.

For accurate higher order derivatives, this Savitzky-Golay Spline Estimation option is algorithm of choice.



Since we are not interested in updating the data table, click on **Cancel** to exit the procedure.

Exit the program by closing the main window or by the **Exit** item in the **File** menu.

Eigendecomposition Filtering

This tutorial covers the signal filtering capabilities of TableCurve 2D using Eigendecomposition and reconstruction techniques.

Generating Test Data



Select the **Generate Data** option in the Edit menu or Main toolbar.

For the Eigendecomposition in this tutorial, we will create a data stream consisting of a sine function, an exponential decay, and white (Gaussian) noise.

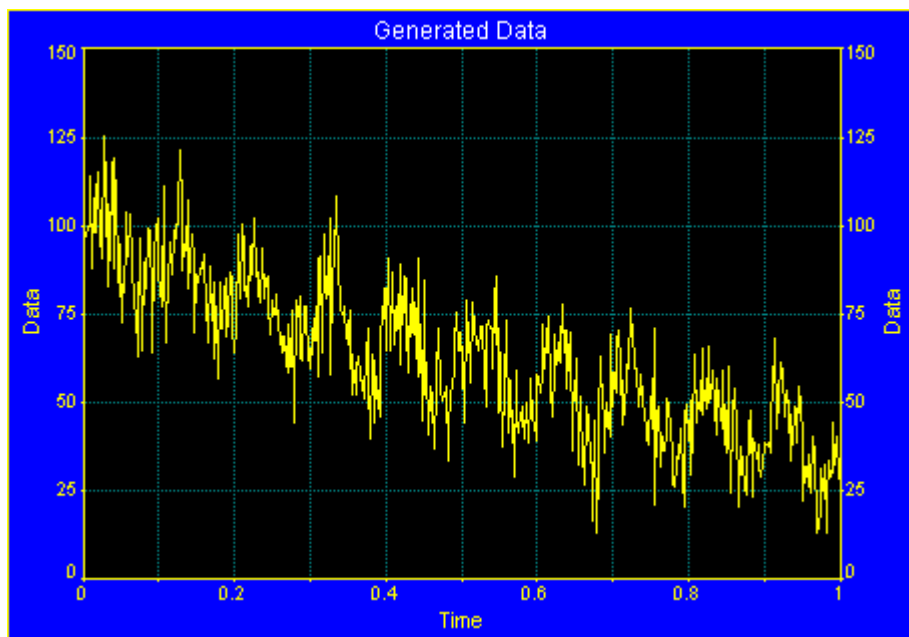
Click **Read** and select the file **tutor1.sig** from the Data subdirectory.

The following signal expression is imported:

```
F1=100*EXP(-X/1)
F2=10*SIN(2*PI*X*10)
Y=F1+F2
```

The X (time) values vary from 0 to 1.0 with a 0.002 sample increment. Gaussian noise is added at a 50% level. The aim is to explore the isolation and recovery of both the sinusoid and the exponential decay.

Click **OK** to process the current signal.



A TableCurve 2D Graph is presented containing the 501 point generated data.



Click **OK** to accept the generated data. Click **Yes** when asked to update the main data table with the revised data.

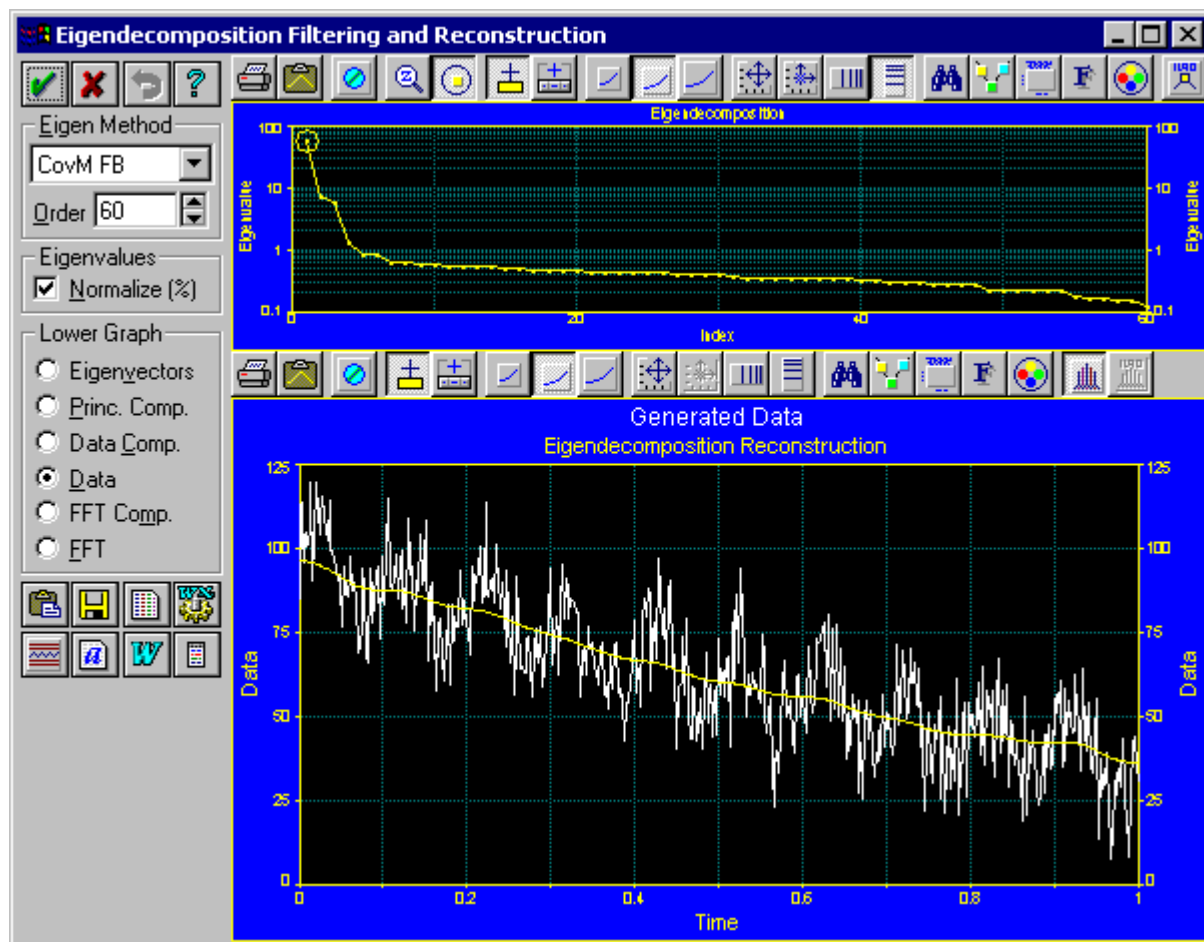
Eigendecomposition Filtration

Unlike Fourier filtering, which isolates components by frequency, Eigendecomposition isolates data

components by signal strength. In order for this test case to work, we had to be certain the first order decay had a greater power at all X-values than the sinusoid.



Select the **Eigendecomposition Filtering** option in the **Filter** menu or main toolbar. Be sure the **CovM FB** algorithm is selected, set the matrix order to **60**. Also be sure that the **Eigenvalues Normalize %** box is checked and that **Data** is checked for the lower graph. Using the left mouse button, box only the first eigenmode in the upper graph.



In an eigendecomposition, a continuously increasing or continuously decreasing data component is captured by a single eigenmode. Note that the first order exponential decay is recovered with virtually no trace of the sinusoid.

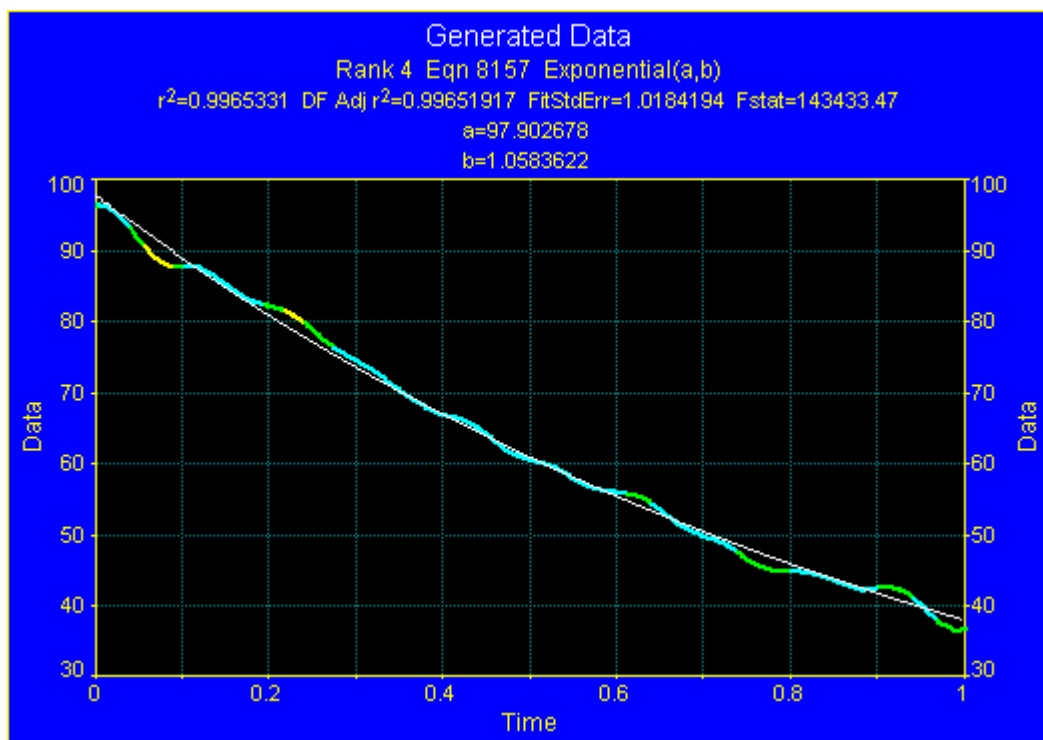


Click on **OK** to close the **Eigenfiltering** procedure and answer **Yes** to update the data table with this first isolated component.



Select the **Curve-Fit Simple Equations** option in the **Process** menu or toolbar. When the fit is concluded, enter the **Review** by clicking the **Graph Start** button.

Click on **Search for Specific Equation** and enter **8157** and click **OK**.



As you will recall, we defined the damped exponential with $F1=100*EXP(-X/1)$. We would thus like to see parameters of (100,1) for the exponential. We are close with (97.9,1.058).



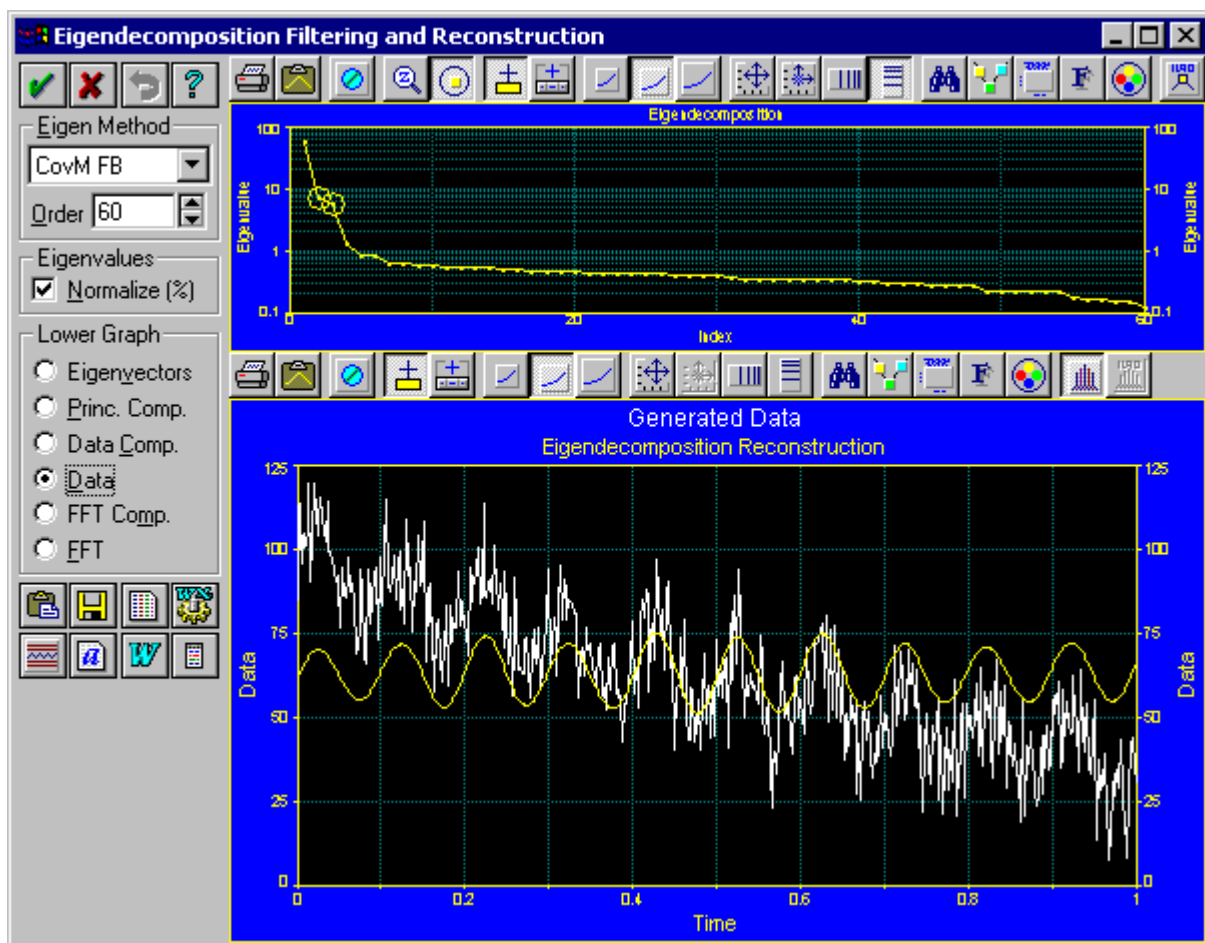
Click OK to close the Review.



Click Reset XY Data to restore the original data.

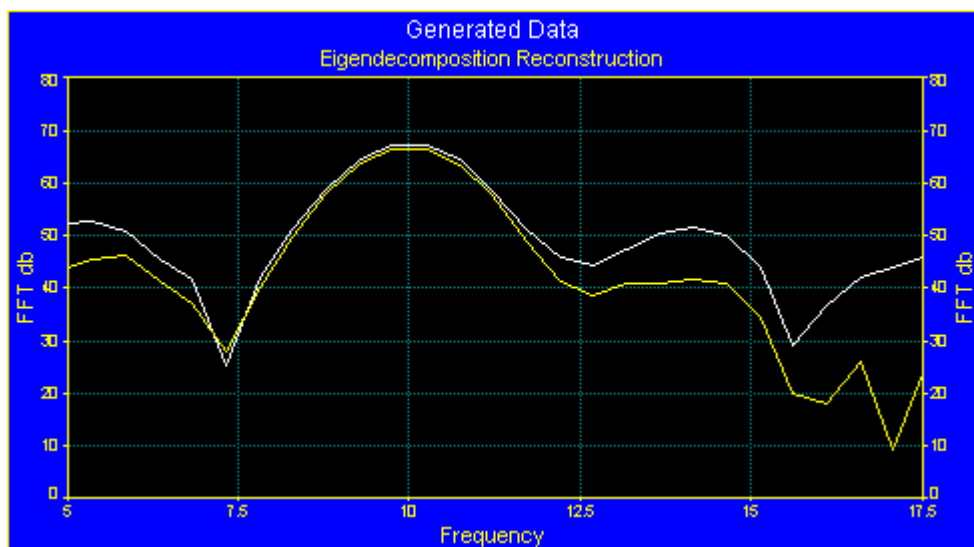


Again select the Eigendecomposition Filtering option in the Filter menu or main toolbar. Using the left mouse button, box the second and third eigenmodes in the upper graph.



Oscillatory components appear as pairs in an eigendecomposition. Note that the sinusoid is recovered with no presence of the exponential decay.

Check the **FFT** box and zoom in the spectral peak in the lower graph.

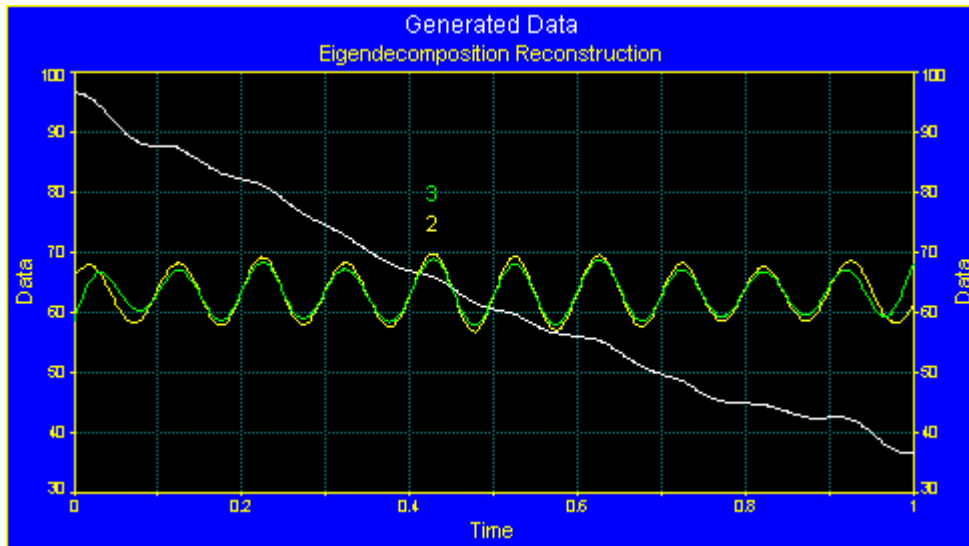


The FFT uses an automatic exact-n FFT procedure with a variable Kaiser-Bessel data taper. Note that the spectral peak's central frequency is at 10. Recall that the sinusoid was generated with $F2=10*\text{SIN}(2*\text{PI}*X*10)$. The sinusoidal frequency was very accurately recovered.



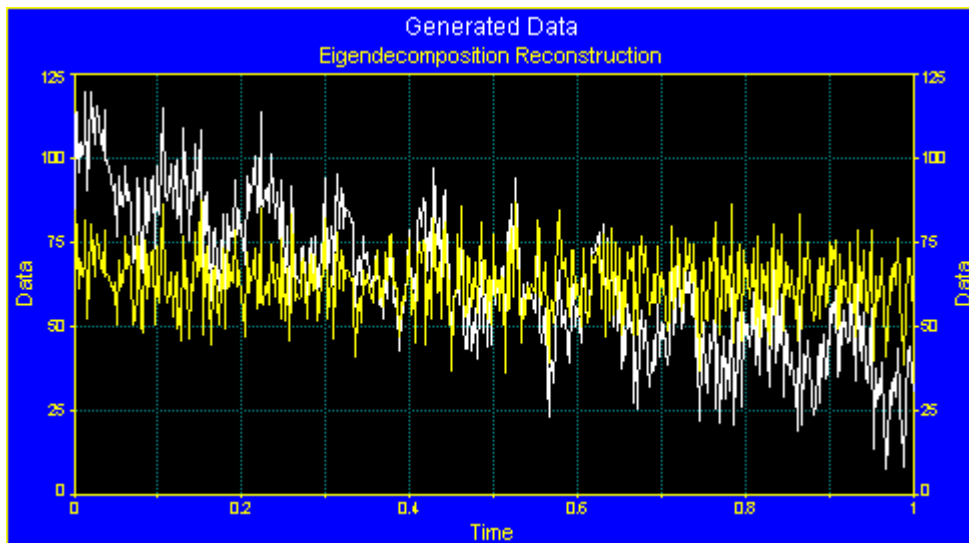
Right click in the lower graph to restore standard scaling or click on **Reset Default Scaling** in the graph's toolbar.

Using the left mouse button, box the first, second and third eigenmodes in the upper graph. Check the **Data Comp.** box.



Inspecting the individual components for the first three eigenmodes reveals the completeness of the separation. Bear in mind that a full 50% level of random noise is present in this data set.

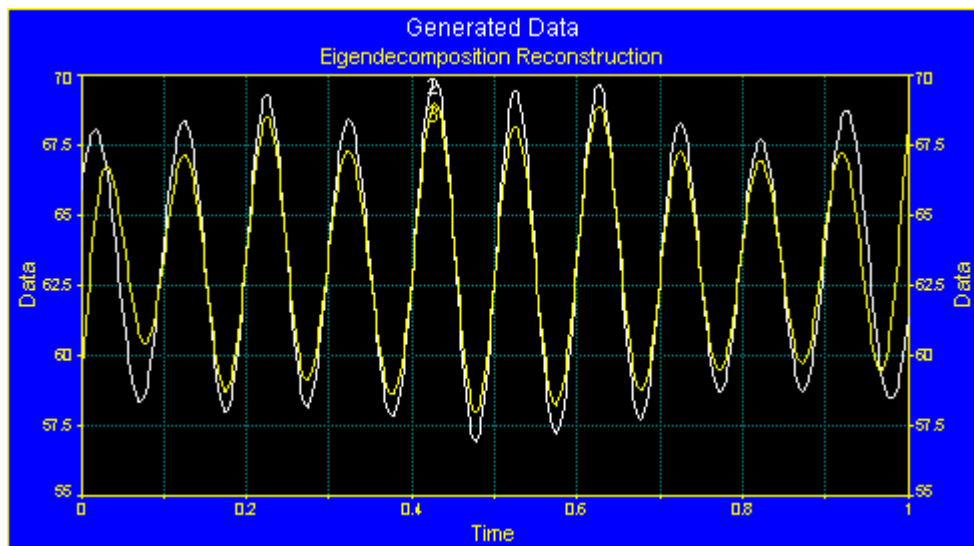
Using the **right** mouse button, again box the first, second and third eigenmodes in the upper graph. Check the **Data** box.



When the right mouse button is used to box the eigenmodes, the enclosed eigenmodes are excluded from the reconstruction. Note that this high level of noise is contained almost entirely in these upper 57

eigenmodes.

Using the left mouse button, again box the second and third eigenmodes in the upper graph. Check the **Data Comp. box**.



These two components sum to produce the reconstructed sinusoid.

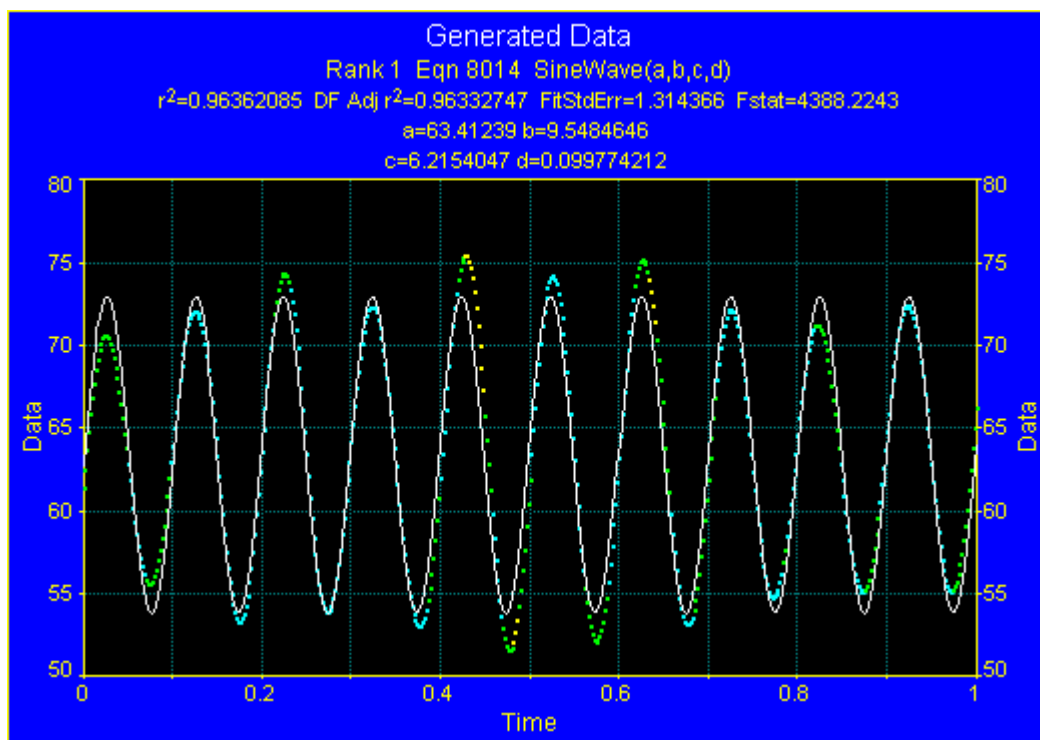


Click on **OK** to close the Eigenfiltering procedure and answer **Yes** to update the data table with this first isolated component.



Select the **Curve-Fit Waveform Functions** option in the **Process** menu or toolbar. When the fit is concluded, enter the **Review** by clicking the **Graph Start** button.

Select equation **8014**, the sine-wave model.



The sinusoid was generated with $F2=10*\sin(2*\pi*X*10)$. The sinusoid's amplitude of 10.0 is recovered as 9.55. The phase of 0 (or 2π) is recovered as 1.978π . The frequency of 10 is recovered as 10.02 (1/wavelength). Although the variations in the sinusoidal amplitude suggest a less than ideal isolation, the sinusoid's parameters are recovered quite effectively.



Click OK to close the Review.



Click Reset XY Data to restore the original data.

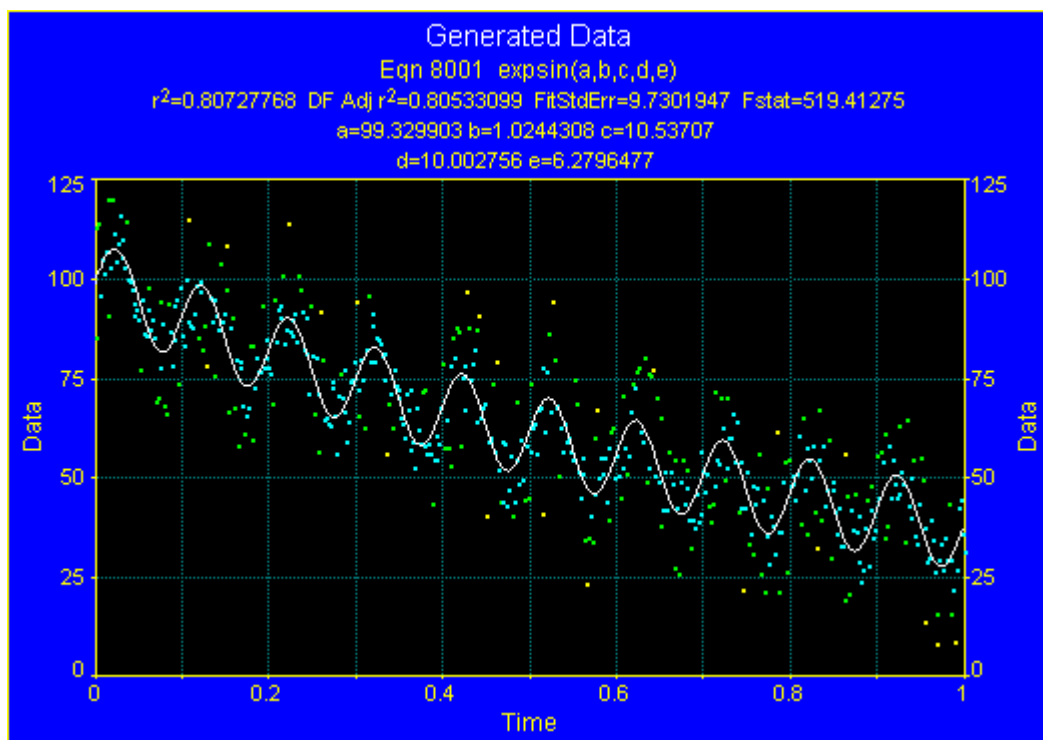
Parametric Fitting



Select the Process menu's User Functions item. In the UDF entry screen, enter **expsin** as the Function Name, 5 as the Adjustable parameter count, $Y=A0*\exp(-X/A1)+A2*\sin(2*\pi*X*A3+A4)$ as the UDF expression, and the values 97.9, 1.06, 9.55, 10.02, and $1.978*\pi$ for the five starting estimates.



Click on the Fit UDFs button. When the fit is concluded, enter the Review by clicking the Graph Start button.



Using the estimates from the two eigenfiltering procedures, the UDF was able to further refine the parameters. Apart from the fact the exponential's amplitude is slightly lower and the sinusoid's amplitude is slightly higher, the generating values are very effectively recovered.



Click **OK** to close the Review.

Exit the program by closing the main window or by the **Exit** item in the File menu.

Fourier Filtering

This tutorial covers the Fourier filtering capabilities of TableCurve 2D.

Generating a Test Signal



Select the **Generate Data** option in the Edit menu or Main toolbar.

For this tutorial, we will create a data set that contains one reference sinusoid and two low power sinusoids, one with a higher and the other with a lower frequency. The low power components will be -40dB and -60dB below the reference sinusoid. White noise will then be added so that a noise floor exists at approximately -75dB. The object of this test is the accurate frequency and power estimation of these two low power components. A secondary focus will be the recovery of phase information for these components.

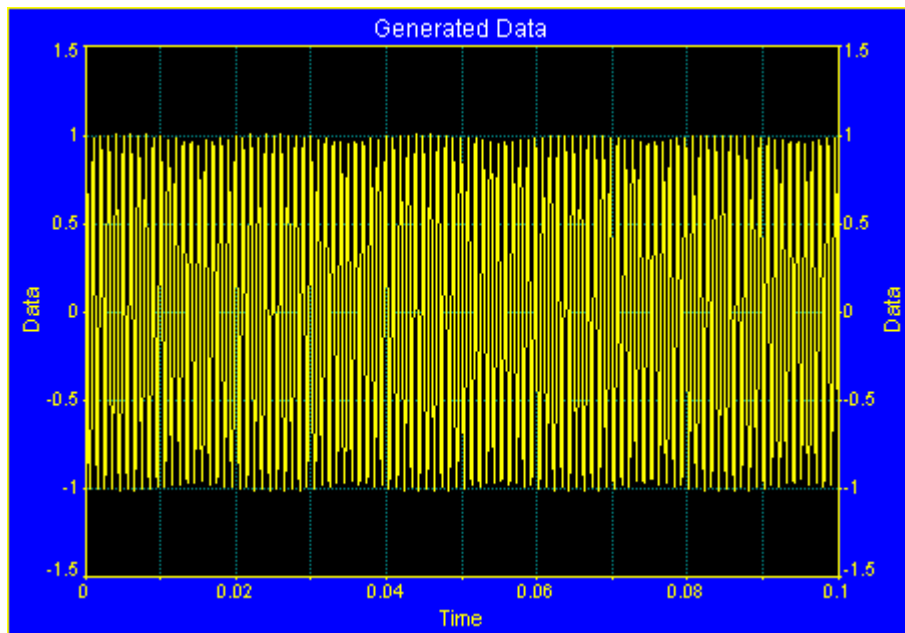
Click **Read** and select the file **tutor2.sig** from the Data subdirectory.

The following signal expression is imported:

```
F1=SIN(2*PI*X*1005+PI/4)
F2=0.01*SIN(2*PI*X*505+PI/2)
F3=0.001*SIN(2*PI*X*1505+PI)
Y=F1+F2+F3
```

The X (time) values vary from 0 to 0.1 with a 0.0001 sample increment. The Nyquist frequency is thus 5000 (half the 10000 sampling frequency). The first low power sinusoid (-40dB) has 1% of the amplitude and 0.01% of the power of one of the reference sinusoid. The other low power test sinusoid (-60dB) has only 0.1% of the amplitude and 0.0001% of the power of the reference sinusoid. The 0.15 Gaussian Noise% creates a white noise floor at about -75dB. It should thus be possible to isolate even the -60dB component given a well designed Fourier filtration.

Click **OK** to process the current signal.





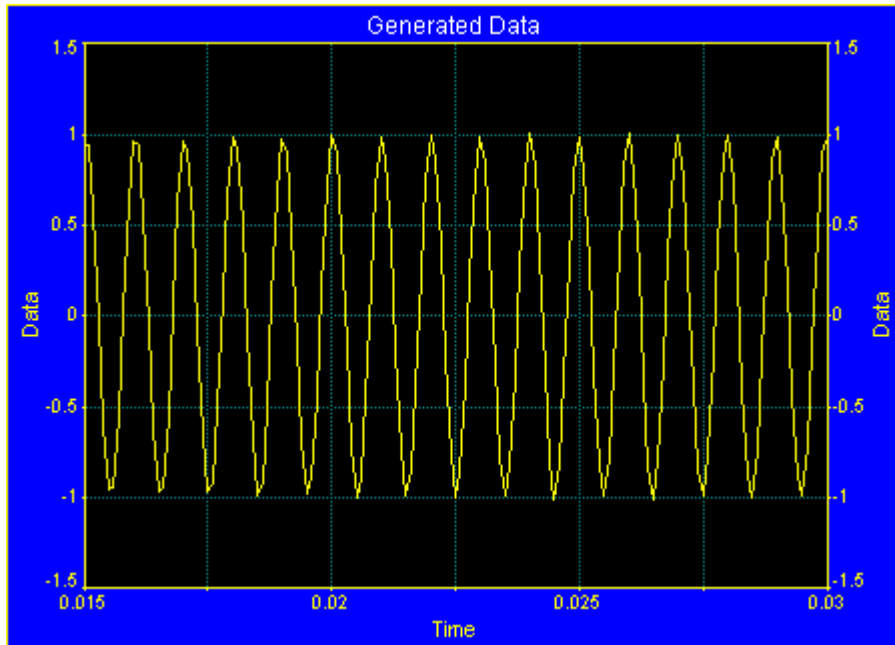
Click **OK** to accept the generated data. Click **Yes** when asked to update the main data table with the revised data. Click **OK** to accept the default titles.

The 1001 point generated data set is now displayed in the main status window. In TableCurve 2D, the main window is informational only. The various analyses and processing take place in separate procedures that are tailored to each specific option.

Viewing the Data



Select the **View Data** option in the **Data** menu or main toolbar. Zoom in on a region of the data.



Inspecting the data visually, only the reference sinusoid is apparent. The two low power sinusoids are imperceptible.

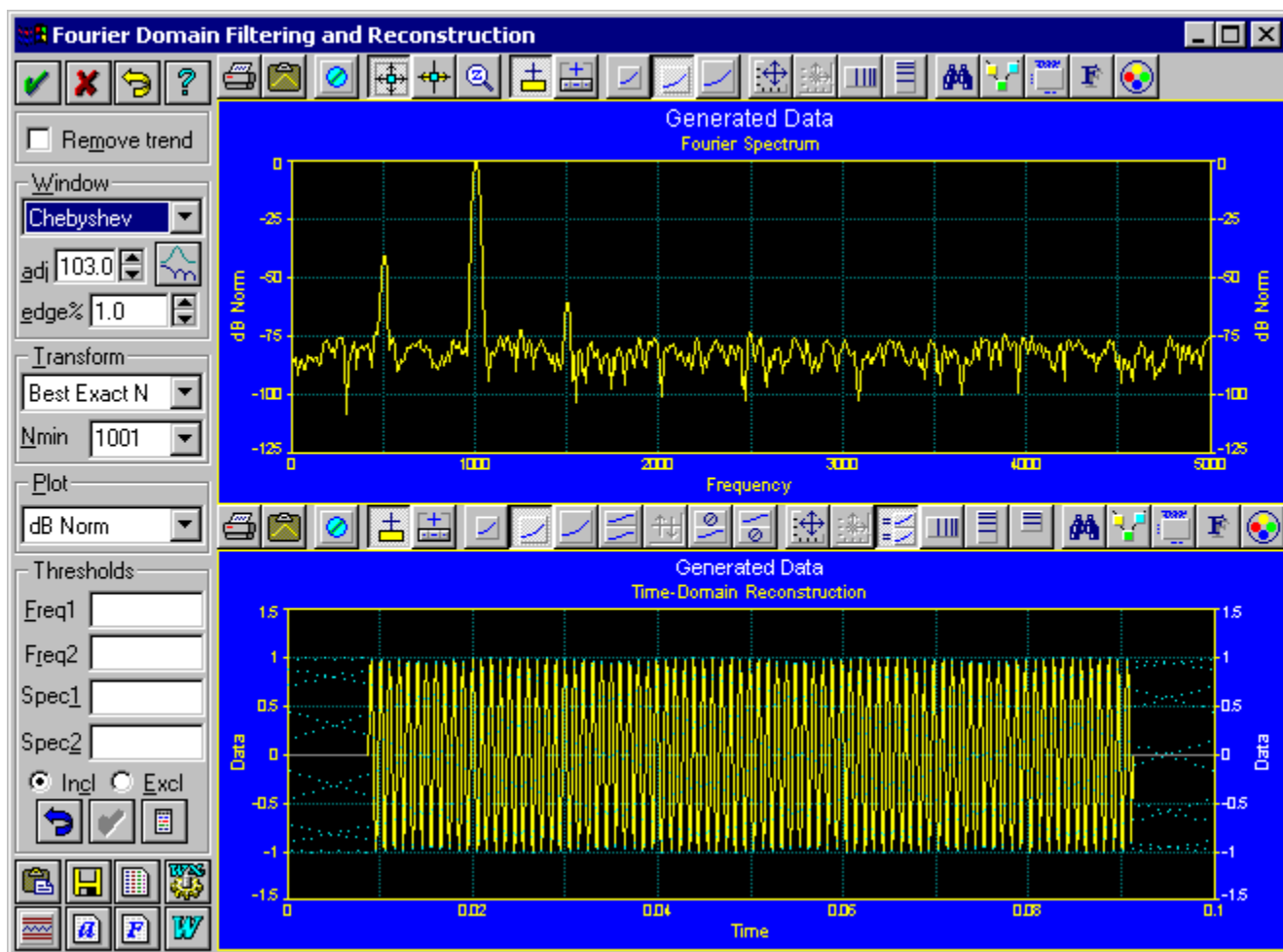


Click **OK** to exit the view option.

Fourier Filtering



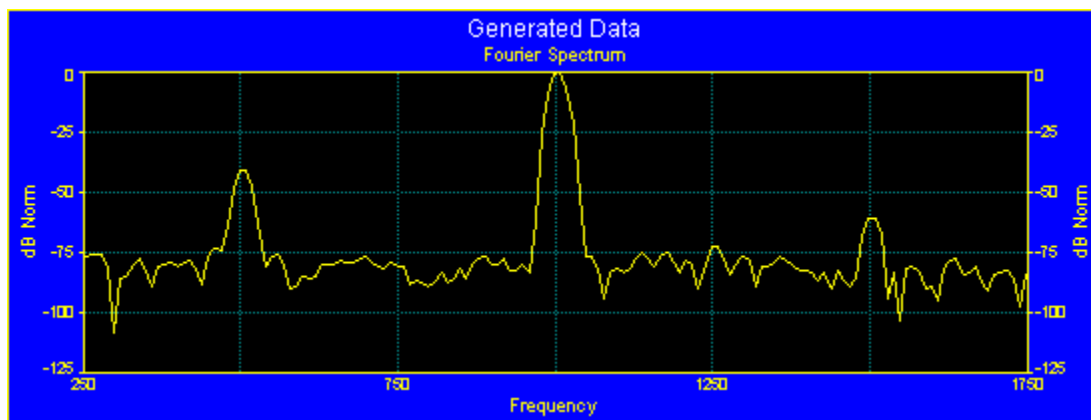
Select the **Fourier Filtering** option in the **Filter** menu or main toolbar. Be sure **Remove trend** is unchecked, set the **Window** to **Chebyshev**, the **adj** parameter to **103**, and the **edge%** to **1.0**. Be sure the default **Best Exact N** algorithm is selected and **Nmin** is **1001**. Select the **dB Norm** plot.



For high dynamic range Fourier spectral analysis, data tapering windows are almost always used. These settings are about optimum for resolving the three components.



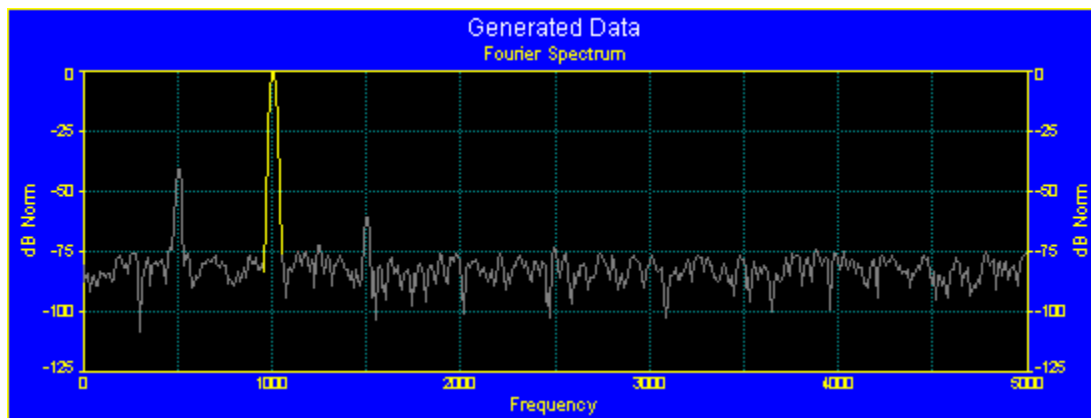
Select zoom-mode and then zoom-in on the three spectral peaks in the upper graph.



Note that not only are the 505, 1005, and 1505 frequencies accurately resolved, but also the 0dB, -40dB, and -60dB spectral powers.



Select XY Sectioning mode. Using the left mouse button, box the largest peak down to about -75dB.

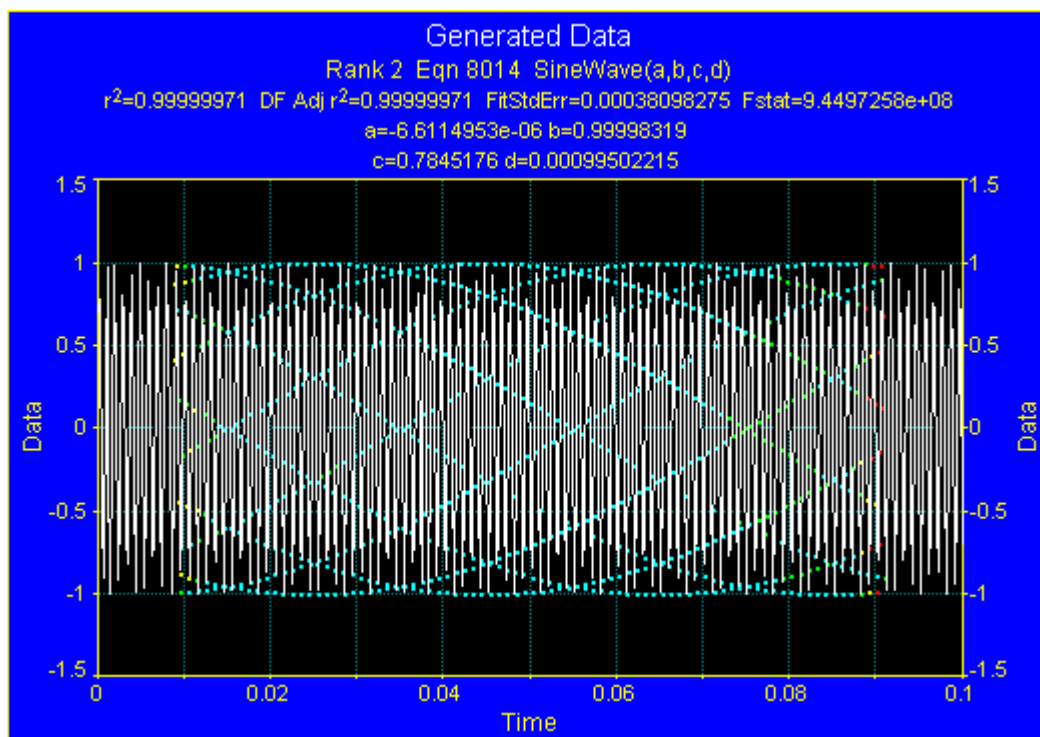


Click on OK to close the Fourier Filtering procedure and answer Yes to update the data table with this first isolated component.



Select the Curve-Fit Waveform Functions option in the Process menu or toolbar. When the fit is concluded, enter the Review by clicking the Graph Start button.

Select equation 8014, the sine-wave model.



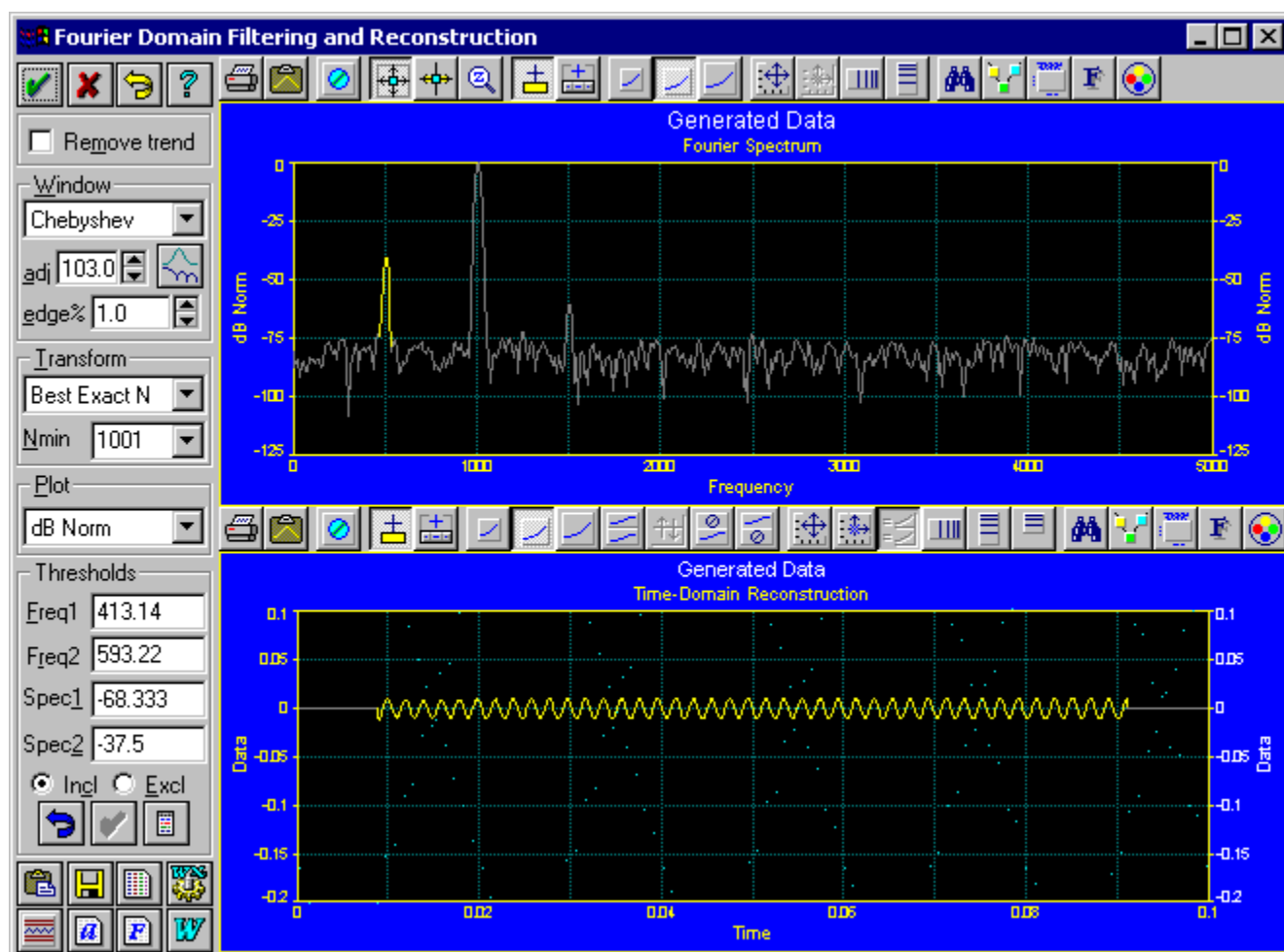
Given this peak's predominance, it is little surprising that the parameters are almost perfectly recovered. The amplitude of 1.0 is recovered as 0.999983, the phase of $\pi/4$ as $\pi/4.0045$, and the frequency of 1005 as $1/0.000995=1005.02$. Note that TC2D's built-in sine model computes a wavelength rather than a frequency.

✓ Click **OK** to exit the Review.

↶ Click on **Reset XY Data** to restore the original test data.

📊 Again select the **Fourier Filtering** option in the Filter menu or main toolbar.

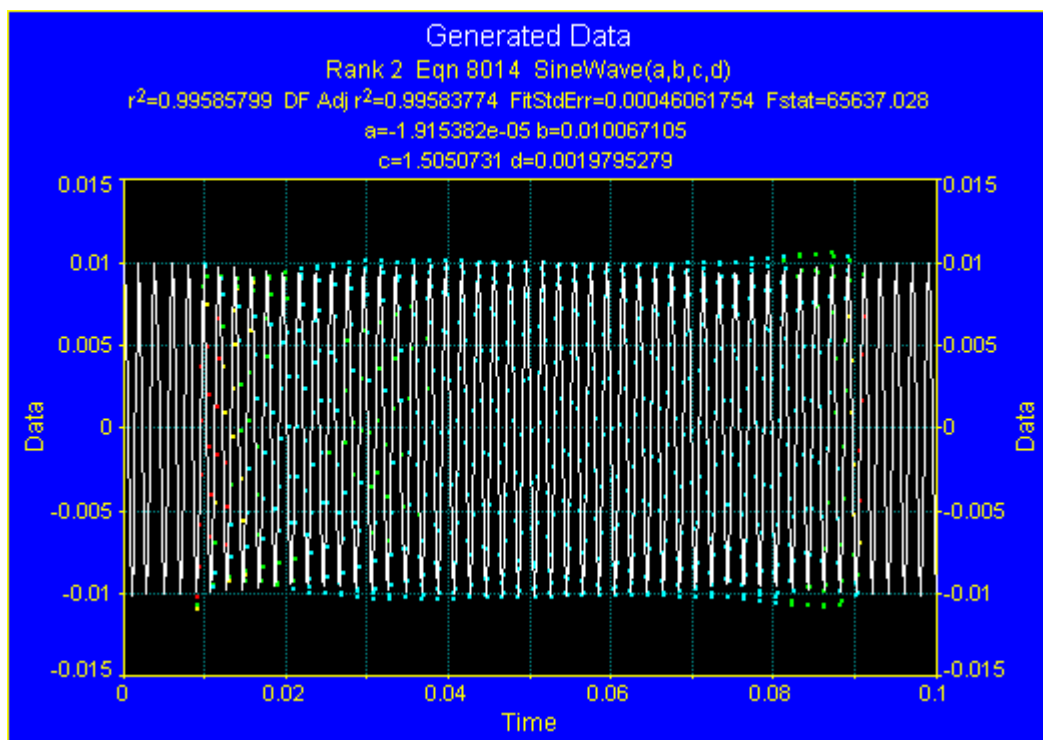
Using the left mouse button, box the lowest frequency peak down to about -75dB. Zoom in on the time domain data in the lower graph.



✓ Click on **OK** to close the Fourier Filtering procedure and answer **Yes** to update the data table with this first isolated component.

📈 Select the **Curve-Fit Waveform Functions** option in the Process menu or toolbar. When the fit is concluded, enter the Review by clicking the **Graph Start** button.

Select equation **8014**, the sine-wave model.



This spectral component has only 1% of the amplitude and 0.01% of the power of the main peak. Still it is recovered with some accuracy. The amplitude of 0.01 is recovered as 0.01001, the phase of $\pi/2$ is recovered as $\pi/2.087$, and the frequency of 505 is recovered as $1/0.001979=505.31$.



Click OK to exit the Review.

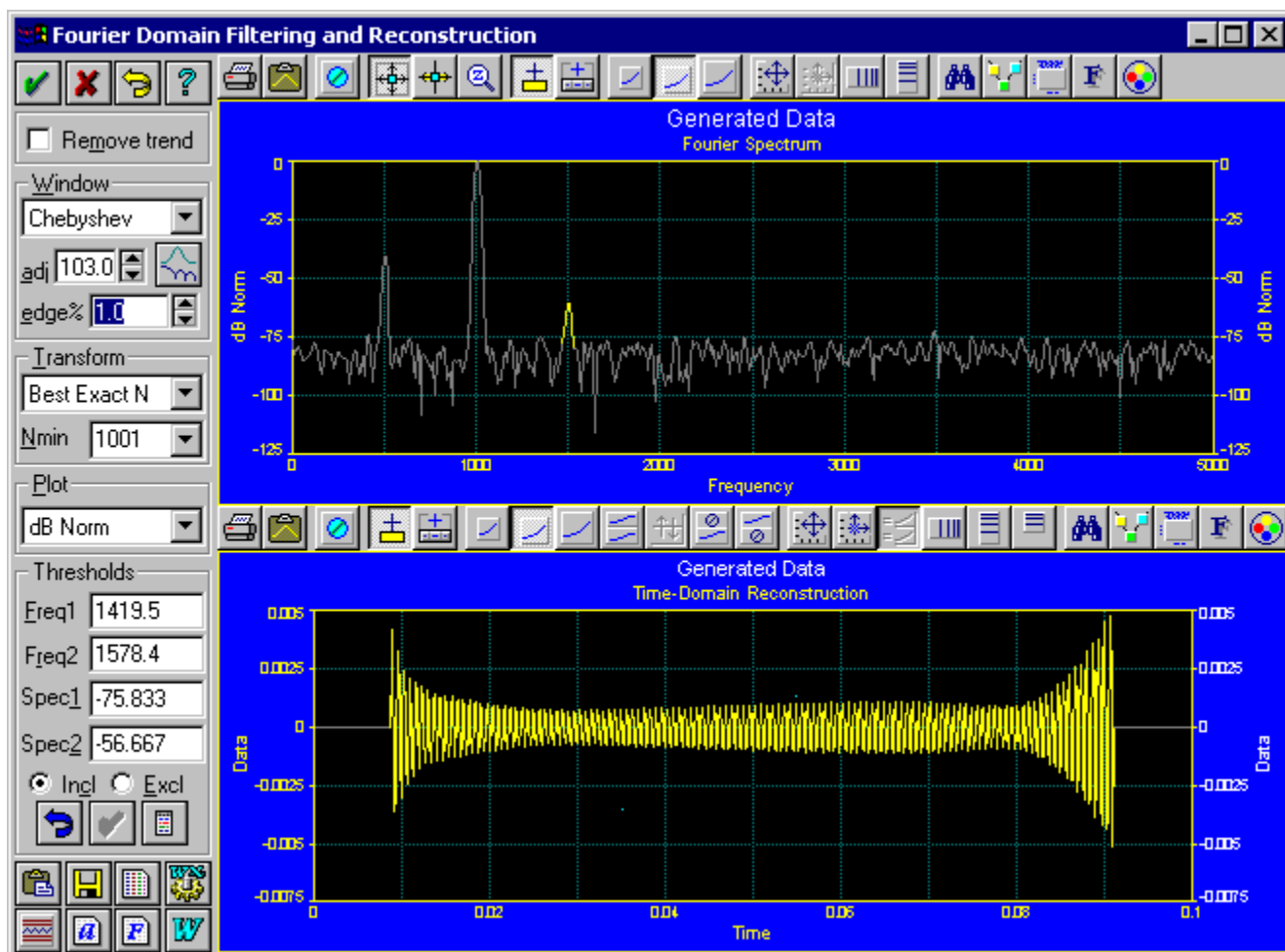


Click on Reset XY Data to restore the original test data.



Again select the Fourier Filtering option in the Filter menu or main toolbar.

Using the left mouse button, box the lowest power peak down to about -75dB. Zoom in on the time domain data in the lower graph.

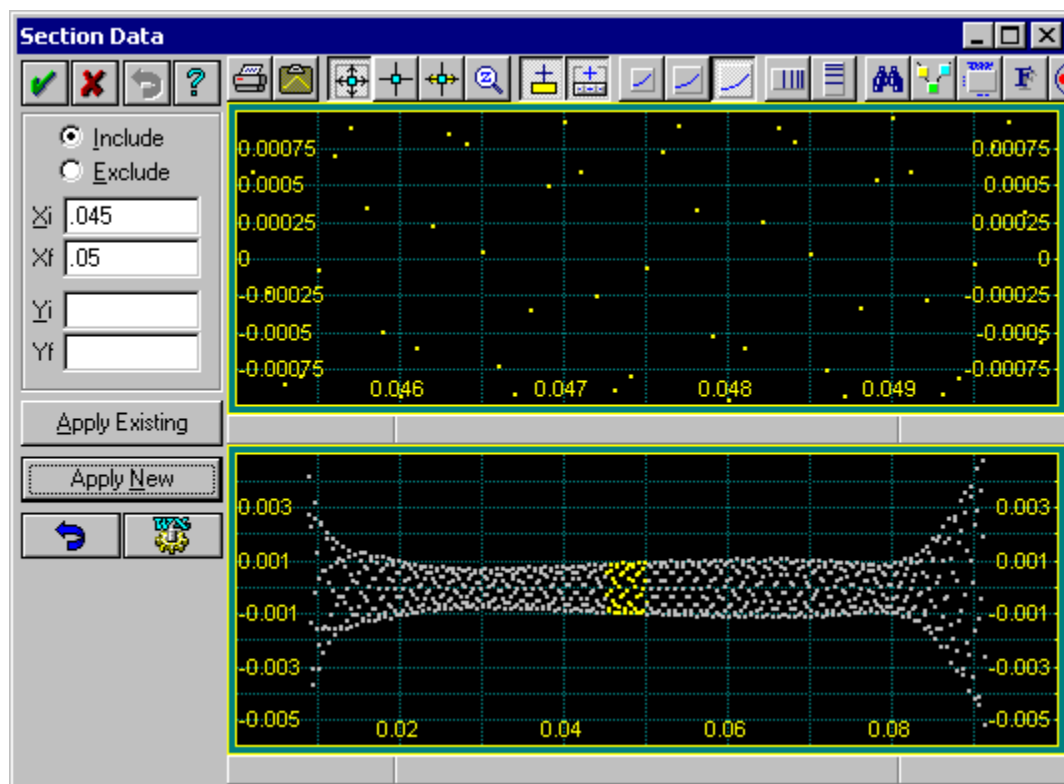


✓ Click on OK to close the Fourier Filtering procedure and answer Yes to update the data table with this first isolated component.

TableCurve 2D currently uses a time domain procedure to scan for estimates of the waveform functions. This procedure works well so long as the number of cycles is not too high. We must thus reduce the number of cycles in the data to be fitted. We must also deal with the anomalies at the edges of the reconstructed data.

✚ Select the Section Data option in the Data menu or main toolbar.

✚ Enter 0.045 for Xi and 0.5 for Xf. Click on Apply New.

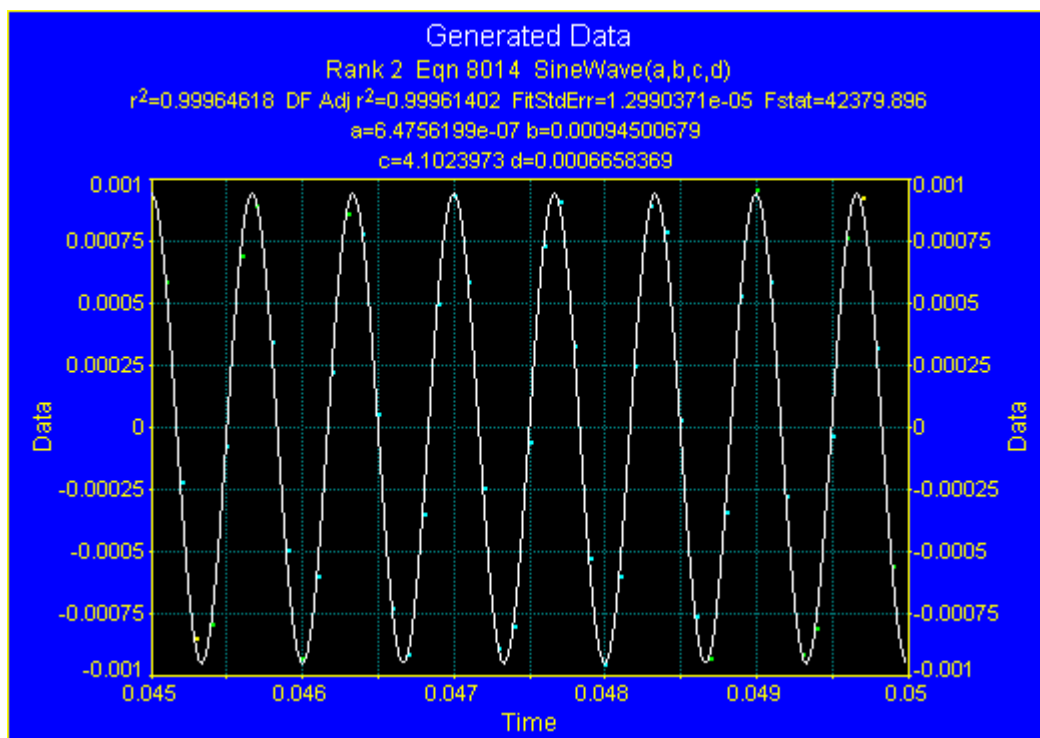


Click on **OK** to accept the sectioning, answer **Yes** to deleting the inactive points, and **No** to saving the old data.



Select the **Curve-Fit Waveform Functions** option in the **Process** menu or toolbar. When the fit is concluded, enter the **Review** by clicking the **Graph Start** button.

Select equation **8014**, the sine-wave model.



This spectral component has only 0.1% of the amplitude and 0.0001% (1 ppm) of the power of the main peak. Despite this very low presence, the sinusoid is adequately recovered. The amplitude of 0.001 is recovered as 0.00095, the phase of π is recovered as $1.31 \cdot \pi$, and the frequency of 1505 is recovered as $1/0.000666=1501.5$.



Click OK to exit the Review.



Click on Reset XY Data to restore the original test data.



Again select the Fourier Filtering option in the Filter menu or main toolbar.

Linear Sinusoidal Fitting

TableCurve 2D v5 offers linear sinusoidal fitting. The peaks within the frequency spectrum are automatically identified, sorted by power, and up to 25 are automatically fitted to a sum of sinusoids model. In linear sinusoidal fitting, the frequencies are fixed at the values determined in the FFT. For a greater accuracy in these frequencies, a bin interpolation procedure is employed. By fitting a sum of sines and cosines, the phase is also linearized. The linear fitting thus computes the amplitude and phase parameters for the sinusoids at the fixed frequencies found in the FFT.



Click on the Numeric Summary button.

The frequency analysis is based directly upon the FFT. It finds the three peaks at 505.1005, and 1505 as well as 22 other spectral features within the noise.

Frequency Analysis

Frequency	Amplitude	Phase	Power	%	Rel %
415.577482	0.00014987	2.41627153	1.1241e-09	4.5002e-06	4.5006e-06
505.098369	0.00712484	1.46921534	2.5407e-06	0.01017120	0.01017225

637.956389	0.00011936	4.09950900	7.1311e-10	2.8548e-06	2.8551e-06
1005.16325	0.70642628	4.77549366	0.02497686	99.9896748	100.000000
1053.48891	0.00015355	3.40435750	1.1801e-09	4.7244e-06	4.7248e-06
1183.81623	0.00011126	3.69921785	6.1951e-10	2.4801e-06	2.4803e-06
1344.91501	0.00012481	3.55058472	7.7969e-10	3.1213e-06	3.1216e-06
1503.78353	0.00066878	3.40847125	2.2385e-08	8.9615e-05	8.9625e-05
1712.07920	0.00010360	3.57119486	5.3715e-10	2.1504e-06	2.1506e-06
2080.27531	0.00013210	2.34054416	8.7343e-10	3.4966e-06	3.497e-06
2179.81897	0.00010370	1.16172654	5.3822e-10	2.1547e-06	2.1549e-06
2256.46209	0.00011913	4.89503962	7.1035e-10	2.8437e-06	2.844e-06
2348.63572	0.00010189	0.51063526	5.1961e-10	2.0801e-06	2.0804e-06
2460.86129	0.00012608	4.46756865	7.9559e-10	3.185e-06	3.1853e-06
2923.33063	0.00010659	3.02230146	5.686e-10	2.2763e-06	2.2765e-06
3124.17822	0.00012836	1.73707390	8.2463e-10	3.3012e-06	3.3016e-06
3303.68903	0.00010463	5.11782383	5.4787e-10	2.1933e-06	2.1935e-06
3484.42147	0.00016586	1.24843320	1.3769e-09	5.5122e-06	5.5127e-06
3696.16562	0.00012169	0.07329151	7.4113e-10	2.967e-06	2.9673e-06
3961.90706	0.00011966	3.45424221	7.1659e-10	2.8687e-06	2.869e-06
4101.96888	0.00012035	3.52112729	7.2495e-10	2.9022e-06	2.9025e-06
4304.92297	0.00010305	1.04096201	5.3145e-10	2.1276e-06	2.1278e-06
4536.10536	0.00010667	0.86048902	5.6951e-10	2.2799e-06	2.2801e-06
4735.07328	0.00010510	1.44313890	5.5287e-10	2.2133e-06	2.2135e-06
4875.93666	0.00010256	1.01889937	5.2645e-10	2.1075e-06	2.1077e-06
			0.02497944	100.000000	

The linear sinusoidal fit will refine the amplitude and phase parameters for these three components.

Sine Component Fit (Suboptimal)

Frequency	Amplitude	Phase	Power	%	Rel %
415.577482	0.00042238	4.25431188	8.9305e-09	1.7871e-05	1.7873e-05
505.098369	0.00976607	1.55126979	4.7697e-06	0.00954500	0.00954601
637.956389	0.00026538	3.24608257	3.5253e-09	7.0547e-06	7.0554e-06
1005.16325	0.99965677	0.73405021	0.04996568	99.9893227	100.000000
1053.48891	0.00309756	1.19783156	4.7982e-07	0.00096019	0.00096029
1183.81623	0.00074628	1.14230882	2.7847e-08	5.5727e-05	5.5733e-05
1344.91501	0.00040205	0.65009692	8.0823e-09	1.6174e-05	1.6176e-05
1503.78353	0.00071390	3.22282557	2.5498e-08	5.1026e-05	5.1031e-05
1712.07920	0.00015652	1.89252228	1.2245e-09	2.4505e-06	2.4507e-06
2080.27531	0.00016562	1.41217869	1.3717e-09	2.745e-06	2.7453e-06
2179.81897	0.00012153	0.52004344	7.3849e-10	1.4778e-06	1.478e-06
2256.46209	0.00011887	5.93346461	7.0614e-10	1.4131e-06	1.4133e-06
2348.63572	9.2722e-05	0.34145593	4.3008e-10	8.6066e-07	8.6075e-07
2460.86129	5.5175e-05	2.75591398	1.5216e-10	3.0451e-07	3.0454e-07
2923.33063	7.5954e-05	1.60422938	2.8838e-10	5.7709e-07	5.7715e-07
3124.17822	6.119e-05	3.70163512	1.8725e-10	3.7471e-07	3.7475e-07
3303.68903	0.00015359	0.86178838	1.1798e-09	2.361e-06	2.3613e-06
3484.42147	0.00012083	1.40643256	7.2996e-10	1.4608e-06	1.4609e-06
3696.16562	0.00010446	5.93396454	5.454e-10	1.0914e-06	1.0915e-06
3961.90706	7.933e-05	3.17425105	3.1463e-10	6.2963e-07	6.2969e-07
4101.96888	8.9498e-05	3.78904193	4.0061e-10	8.0169e-07	8.0178e-07
4304.92297	0.00011298	0.91802213	6.3819e-10	1.2771e-06	1.2773e-06
4536.10536	8.8996e-05	1.12938285	3.961e-10	7.9267e-07	7.9275e-07
4735.07328	0.00021892	1.50943628	2.3963e-09	4.7953e-06	4.7958e-06
4875.93666	9.4486e-05	0.33286241	4.4636e-10	8.9323e-07	8.9333e-07
			0.04997102	100.000000	

r^2	DOF r^2	Std Err	F-stat
0.9991292255	0.9990586221	0.0216956823	14358.041721
Data Power	Model Power	Error Power	Ratio
0.0500555104	0.0500123879	4.358706e-05	1147.4135605

We will extract these three components for starting estimates in a non-linear 3-sine UDF that we will now fit.

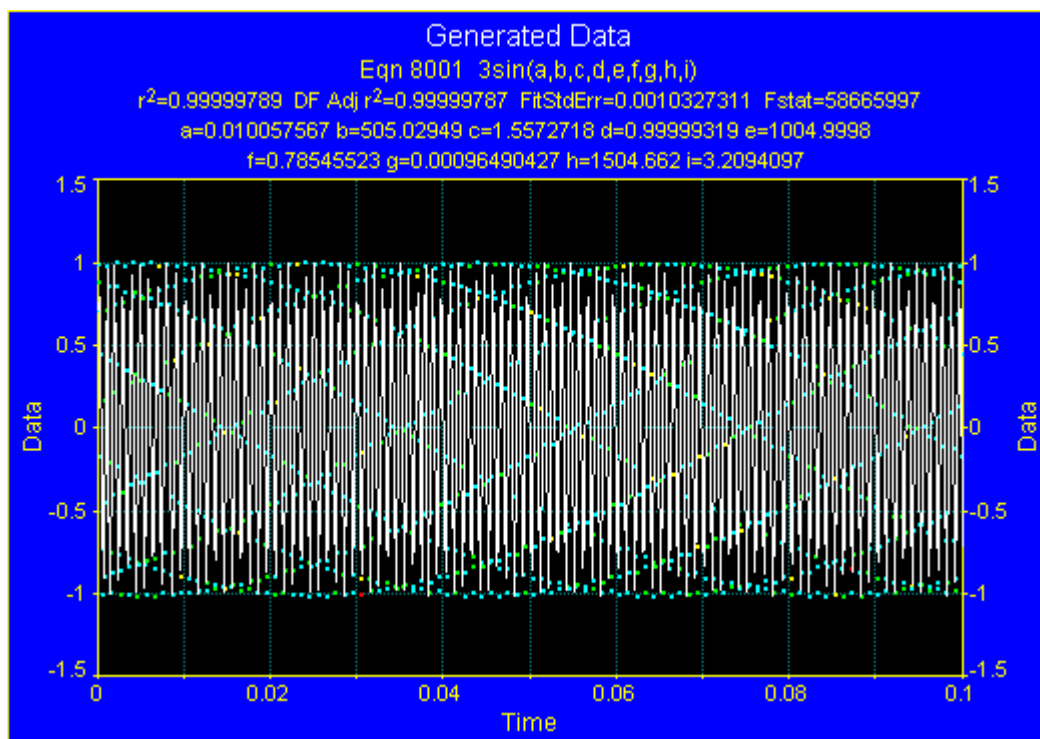
Frequency	Amplitude	Phase
505.098369	0.00976607	1.55126979
1005.16325	0.99965677	0.73405021
1503.78353	0.00071390	3.22282557

Close the Numeric Summary and then Click on **OK** to close the Fourier Filtering procedure.

Parametric Fitting

 Select the Process menu's **User Functions** item. In the UDF entry screen, enter **3sin** as the Function Name, **9** as the Adjustable parameter count, **$Y=A0*\sin(2*PI*X*A1+A2)+A3*\sin(2*PI*X*A4+A5)+A6*\sin(2*PI*X*A7+A8)$** as the UDF expression, and the values **0.0098, 505.1, 1.55, 1.0, 1005.2, 0.73, .00072, 1503.8, and 3.22** for the nine starting estimates.

 **Fit UDFs** Click on the **Fit UDFs** button. When the fit is concluded, enter the Review by clicking the **Graph Start** button.



Using the estimates from the linear sinusoidal fitting, the UDF was able to refine the parameters of the sinusoids:

- All parameters in the high power component (1,1005,0.7864) are recovered to 5-6 decimal places.
- The first low power spectral component (.01,505,1.5707) is recovered as (.01001,505.03, 1.5572).
- The lowest power spectral component (.001,1505,3.1416) is recovered as (.00096,1504.66,3.2094)



Click **OK** to close the Review.

Exit the program by closing the main window or by the **Exit** item in the File menu.

The Fourier filtering capabilities in TableCurve 2D are drawn from the signal analysis software product, AutoSignal. AutoSignal offers automated harmonic analysis and non-linear optimization with no practical limit on the number of spectral components.

Prediction and Forecasting

This tutorial covers the prediction capabilities of TableCurve 2D.

A **prediction** procedure applies to data with equally spaced x-values and generates estimates additional discrete x values, at one or both ends of a data stream, as in time-series forecasting. No interpolation occurs and there is no continuous function, although it is possible to estimate or predict an arbitrary number of additional data values with this same x-spacing.

An **estimation** procedure, on the other hand, is one where the set of x values used for the computation of y values is arbitrary and user-defined. Interpolation is always possible, and as with curve-fit equations, extrapolation or prediction is available, though not always wise. A continuous function, or the means to simulate such, is the central feature of the procedure.

TableCurve 2D offers a single prediction procedure based upon autoregressive modeling.

AR Linear Prediction

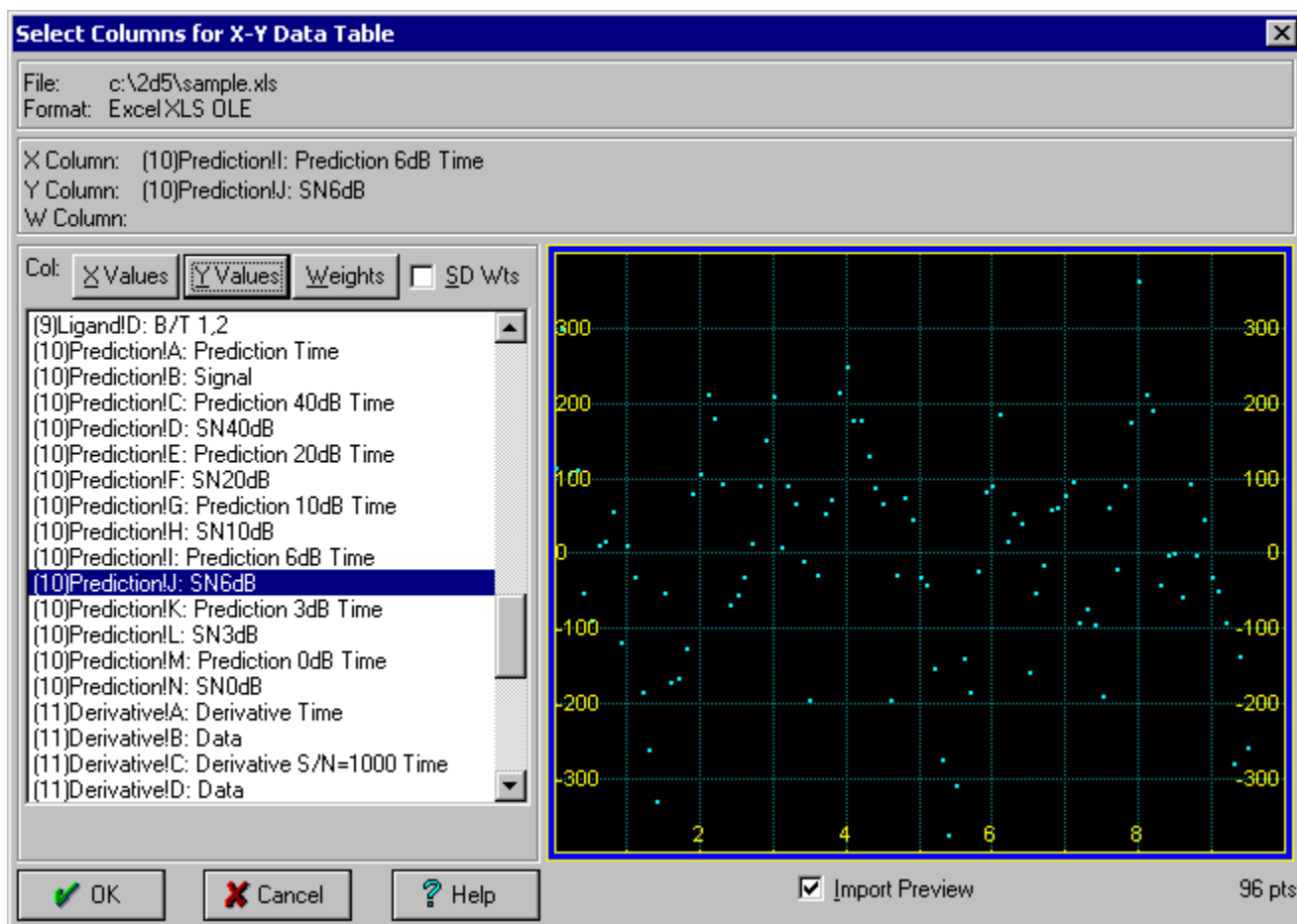
An AR (Autoregressive) model forecasts future values by a linear relationship with some number of prior values. It can also be used to back forecast prior values by a linear relationship with some number of subsequent values. In this example, we will only explore the prediction of future values.

Start TableCurve 2D. Select Start/Programs/TableCurve 2D v5.



Select the File menu's Import option (or use the Import button in the main toolbar). If Excel [xls] files are not shown, click on the Files of Type drop-down button and select Excel [xls] files. Select and open the file SAMPLE.XLS.

Select the column with the label (10)Prediction!I: Prediction 6dB Time to be used as the X-variable in the data table. Select the next column identified in the selection list as (10)Prediction!J: SN6dB for the Y-variable. Check Import Preview to see a graph of the data that will be imported.



Note that the first two selections are automatically placed in the X and Y positions. You may double click on the column for the Y variable in order to immediately proceed with the read operation. You can also revise the initial X,Y selections as well as to specify a column to be used for the weights. The weights can be optionally imported as standard deviations.

Press OK to accept these choices. Press OK once again within the titles dialog to confirm the imported titles.

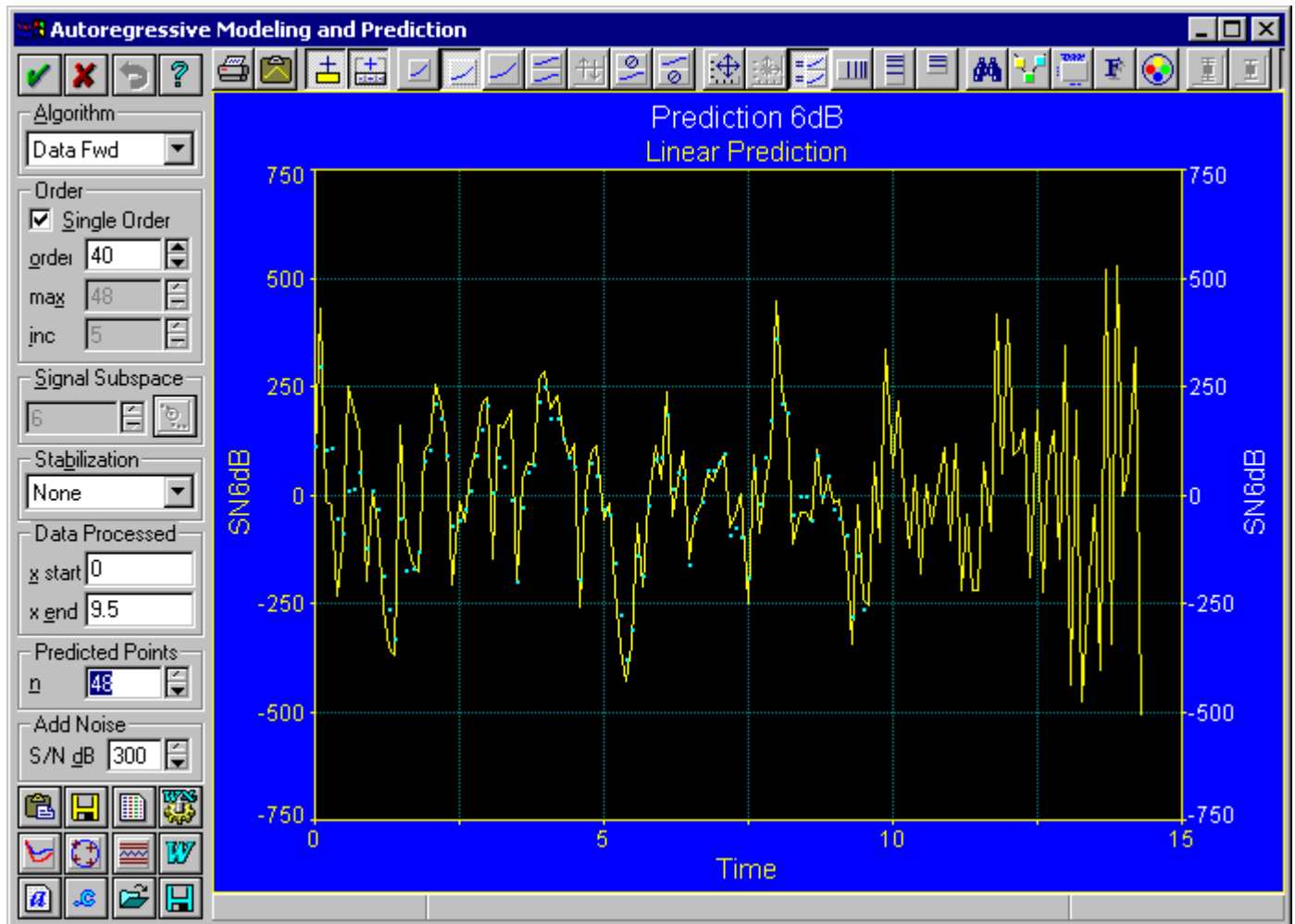
This data set consists of the sum of three different sinusoids and 50% Gaussian noise.

Autoregressive Fitting



Select the **AR Modeling and Prediction** option in the Estimate menu or Process toolbar.

Select the **Data Fwd** algorithm, be sure **Single Order** is checked, and that the order is set to **40**. Be sure **Stabilization** is set to **None**. Set the Predicted Points **n** to **48**.



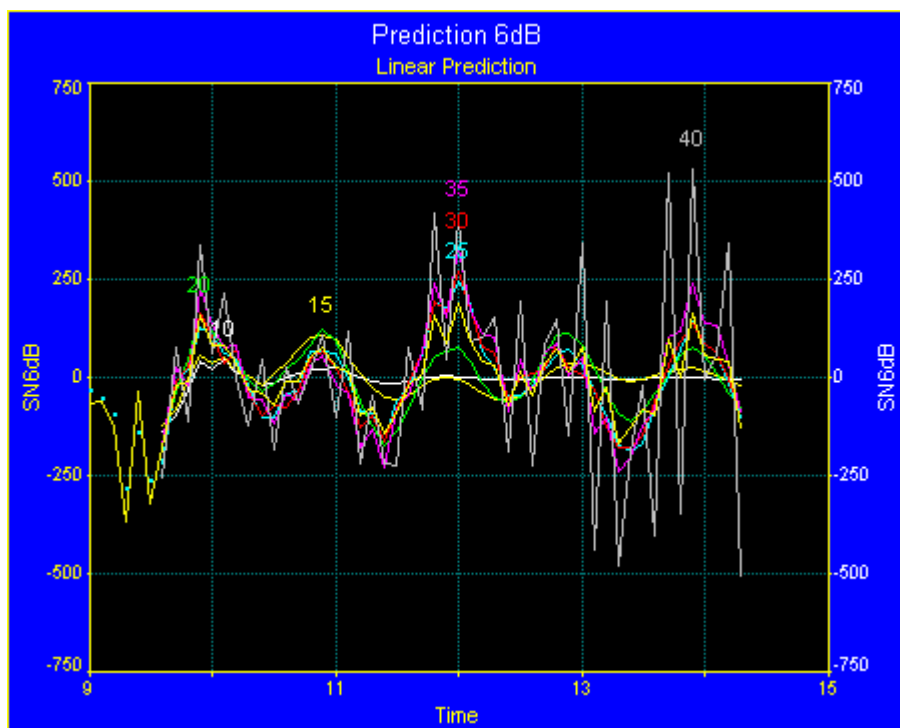
The **Data Fwd** algorithm is really one of the best AR methods for prediction since it performs a least-squares minimization when it computes the coefficients. The AR procedures in many software packages employ reflection coefficient or autocorrelation based algorithms which produce results that are even less credible.

Still, it would not be much of a reach to assume that this prediction is nonsense. But is it?

Ascertaining Nonsense

This is one of the most important elements in using AR Prediction wisely. One of the best ways to assess the validity of a prediction is to see if the same trends are found at a variety of model orders. TableCurve 2D makes this easy.

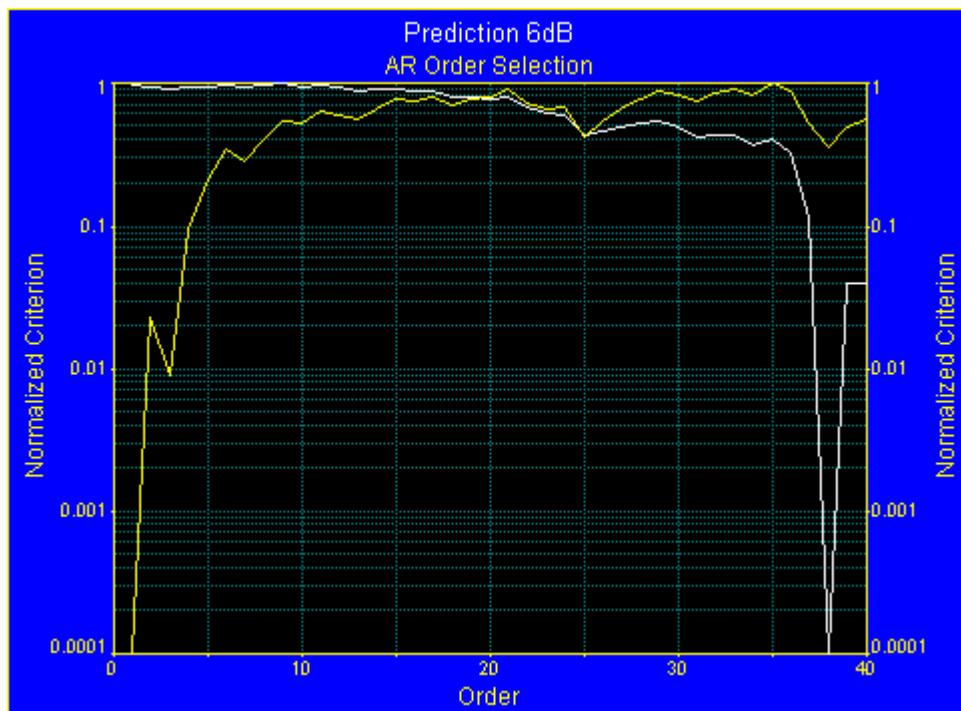
Uncheck Single Order. Enter **10** for min, **40** for max, and **5** for inc. Zoom in on only the predicted portion of the graph.



There is an indication of common upward and downward trends at certain values of time, suggesting some validity to the prediction. It is well known, however, that AR predictions are very sensitive to model order. So much is this expected with traditional AR modeling that an extensive measure of research has gone into developing various numeric criteria for identifying the optimum model order.

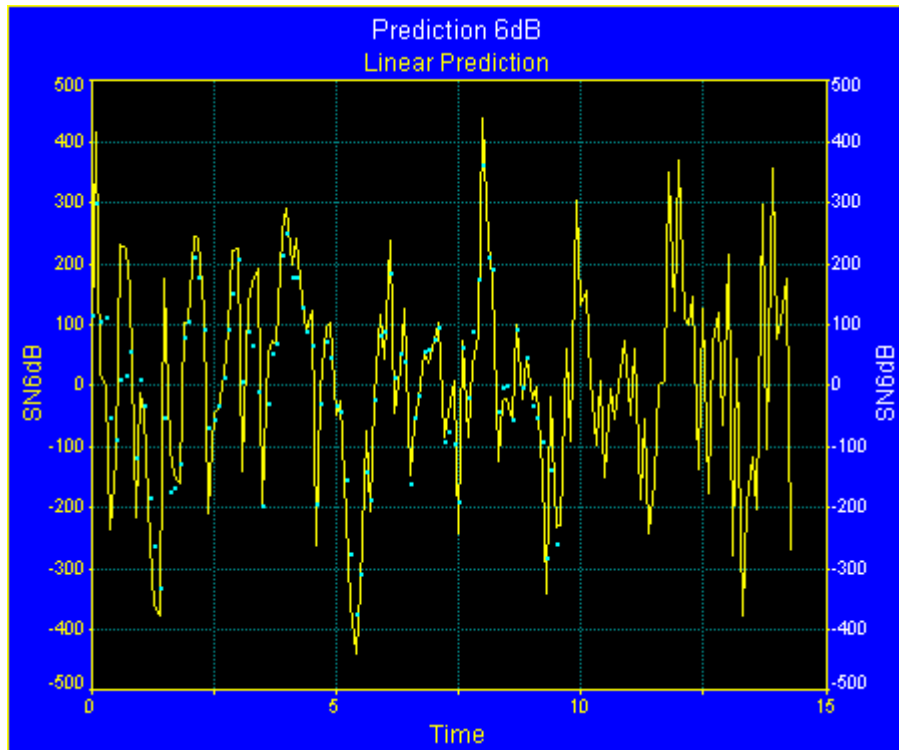


[Click on Plot Selection Criteria.](#)



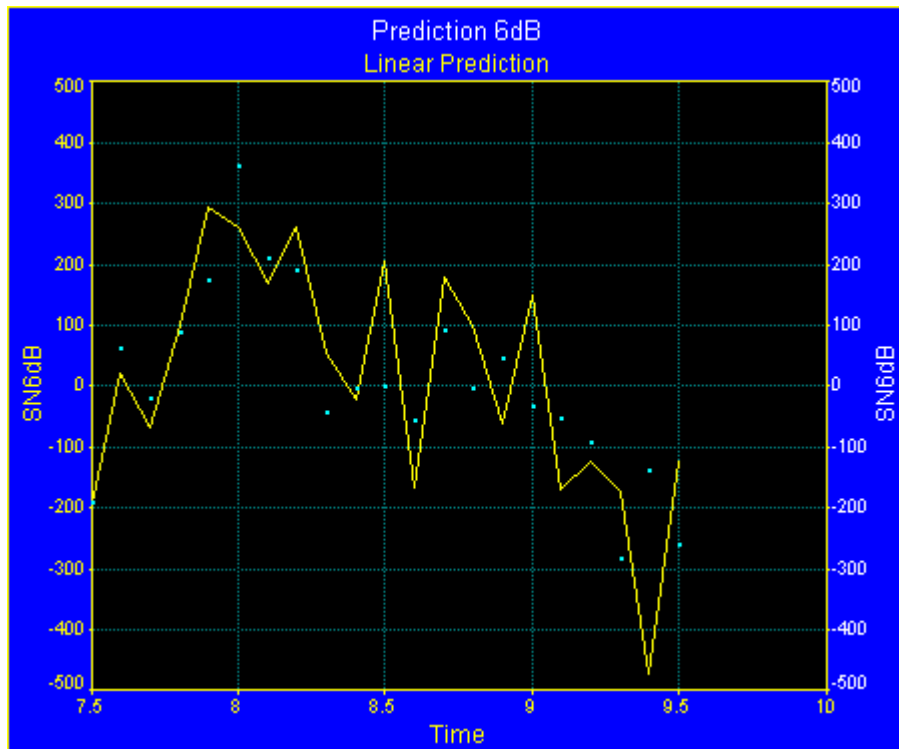
The white curve is a normalized AIC (Akaike Information Criterion) and the yellow curve is a normalized MDL (Minimum Description Length). The minimum in these criteria are possible optimum model orders. Both criteria suggest 38 as the optimum model order.

Check Single Order. Enter 38 for the order. Right click in the graph area to restore standard scaling.



The order 38 prediction does appear slightly more stable. One of the best ways to check for validity with any given AR fit is to use only a portion of the data series to compute the AR model and use the remainder as a check for accuracy in the prediction.

In the Data Processed box, enter 7.8 for x end and set the Predicted Points n to 17. Zoom in on the predicted region.



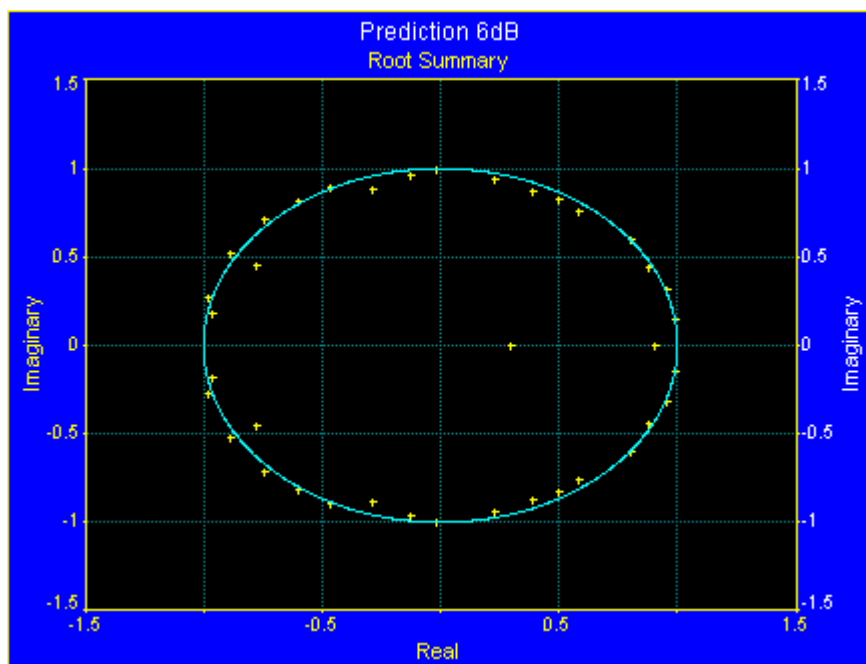
It is hard to argue with what is evident here. The prediction does roughly track the data. In this case, the prediction cannot be dismissed as nonsense.

Assessing Stability

A set of AR coefficients is a filter that is applied to the end of the data to generate the future values. It is important to verify that the coefficients are stable. The complex roots of an AR polynomial must fall on or within the unit circle in order to have a stable filter.

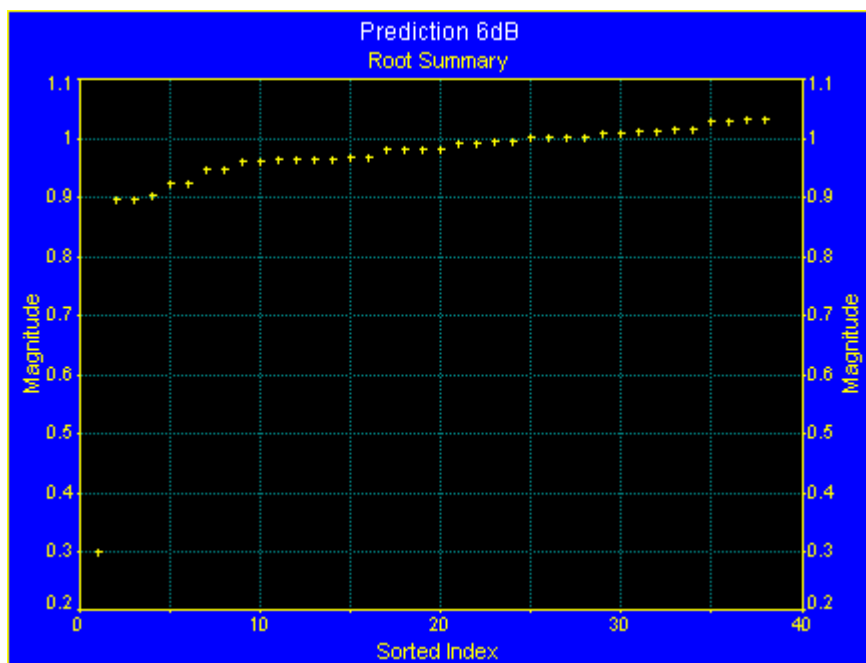


Click the **Plot Roots** button.



Unlike the autocorrelation and reflection coefficient AR algorithms, a least-squares minimization does not insure that these complex roots will be in or on the unit circle. Here they are close, but note the warning that 14 of these roots are outside the unit circle.

Check the Magnitude box.



The roots with a magnitude greater than 1.0 are outside the unit circle. Here it is evident that a significant number of roots are outside and contributing to an instability in AR predictions.

In harmonic analysis, the inspection of complex roots can help identify the number of harmonics present.

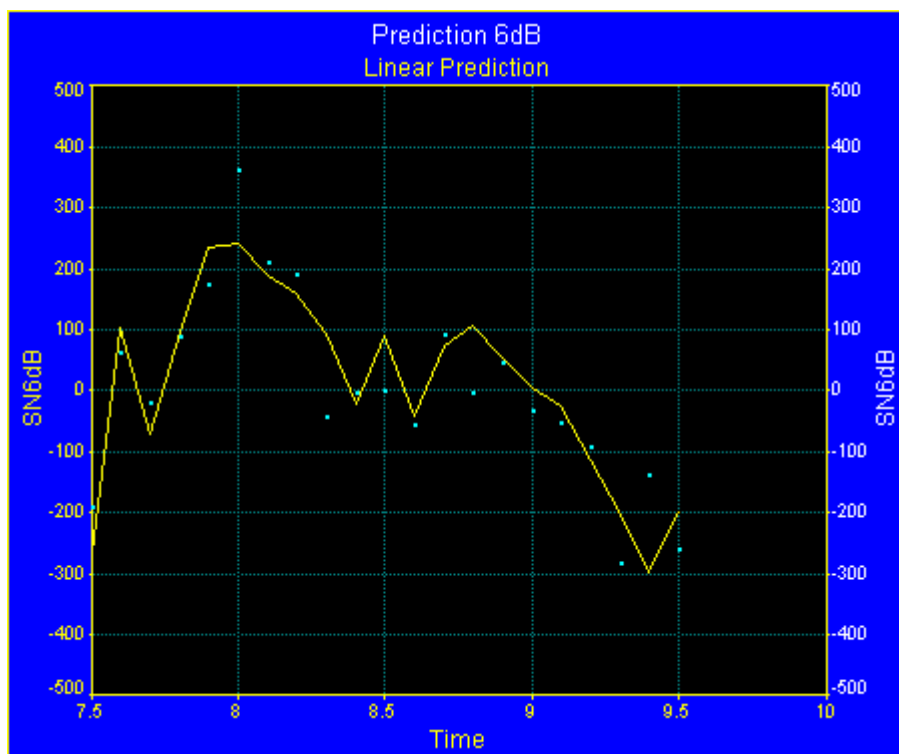
Roots very near the unit circle are typically associated with signal harmonics and those in the interior are generally attributable to noise. This data set was generated by summing three sinusoids and adding a large measure of noise. There should thus be six roots very close to the unit circle and the remainder should be in the interior. Clearly that is not evident here.

Stabilizing an AR Fit

The set of AR coefficients can be stabilized in one of two ways. The roots can be reflected back into the interior of the unit circle. This is the choice to make if the roots most removed from the unit circle represent noise. They can also be mapped directly to the unit circle. This is the option to use if these roots furthest from the unit circle represent harmonics. The easiest approach is to try both stabilizations and see which is most effective.

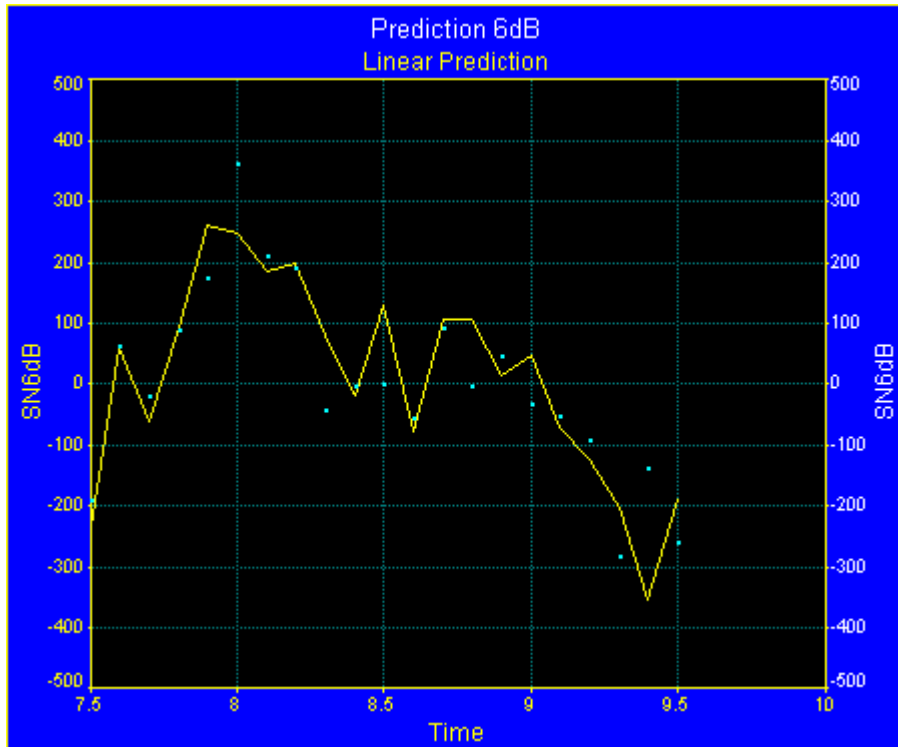


Click OK to close the Complex Roots plot. For the Stabilization, select Reflect Out.



There is a lot to like about this stabilization which assumes the outlier roots are due to noise.

For the Stabilization, select Unit Circle Out.



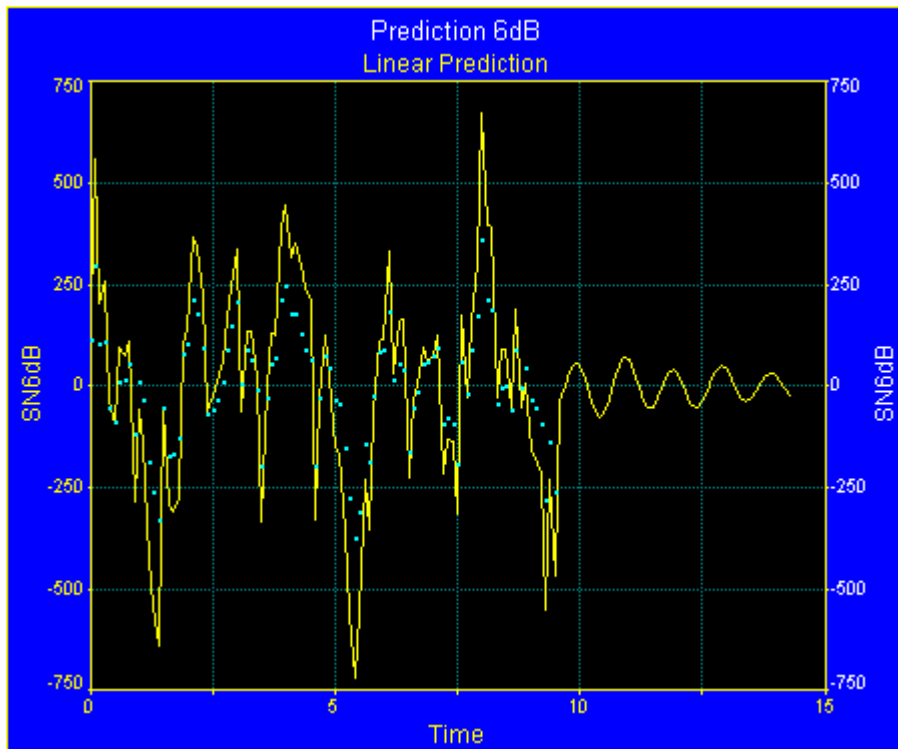
Reflecting the outlier roots to the unit circle, assuming they represent harmonics, also results in an improved prediction.

SVD Autoregressive Prediction

While we appear to have a respectable prediction, can we do better? Indeed, we can. It is possible to compute the AR coefficients using SVD (singular value decomposition) to remove the influence of noise on the AR coefficients. In this case, the model order is far less important, since most of the eigenmodes are going to be discarded anyway.

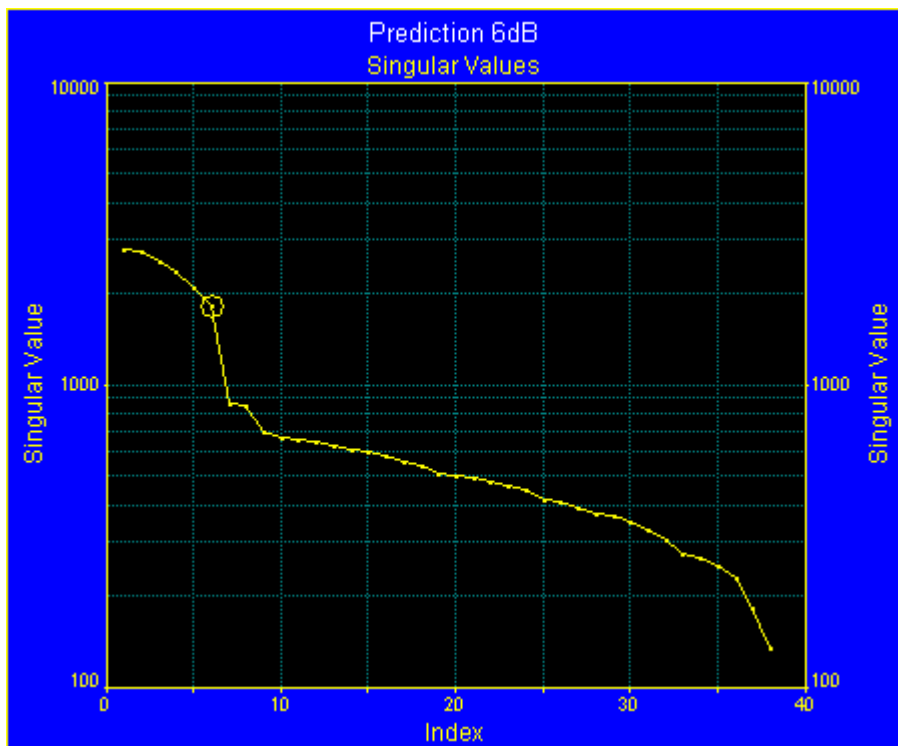
It is important to fit a high enough model order to manage the noise that is present, and it is crucial that a signal space be selected. Unlike the optimum model order identification for traditional AR fitting, which is often tricky, signal space identification is often straightforward.

Right click in the graph area to restore standard scaling. Select the Data Svd Fwd algorithm. Leave the order at 38 and set the Signal Subspace to 2. Set the Stabilization to None. Set x end to 9.5 and set the Predicted n points to 48.



A signal space of 2 accommodates only a single harmonic. We know that this data set was constructed from three sinusoids and that the correct signal space should be 6.

Click on the Graphically Select Signal and Noise Sub-Spaces button. Click on the 6th eigenmode.

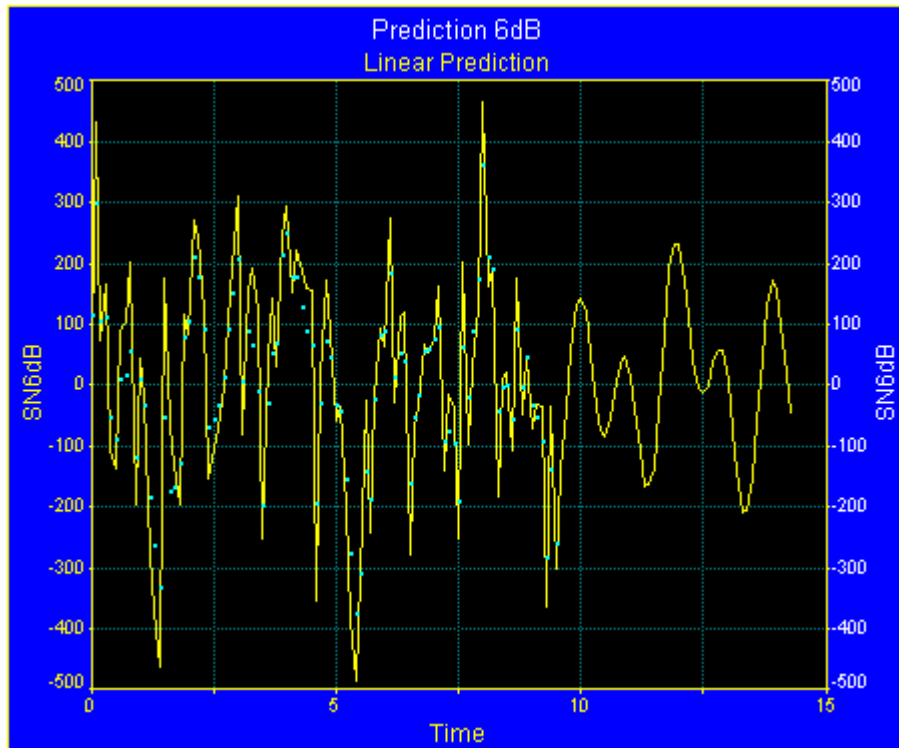


A singular value plot is often sufficient for setting the signal space. This is the eigenmode count used for computing the AR coefficients. Ideally, this plot will reveal a transition to a long sloping noise floor. The last eigenmode representing signal space should be just prior to this transition.

In this plot there are two possible signal spaces, 6 and 8. We will explore each.

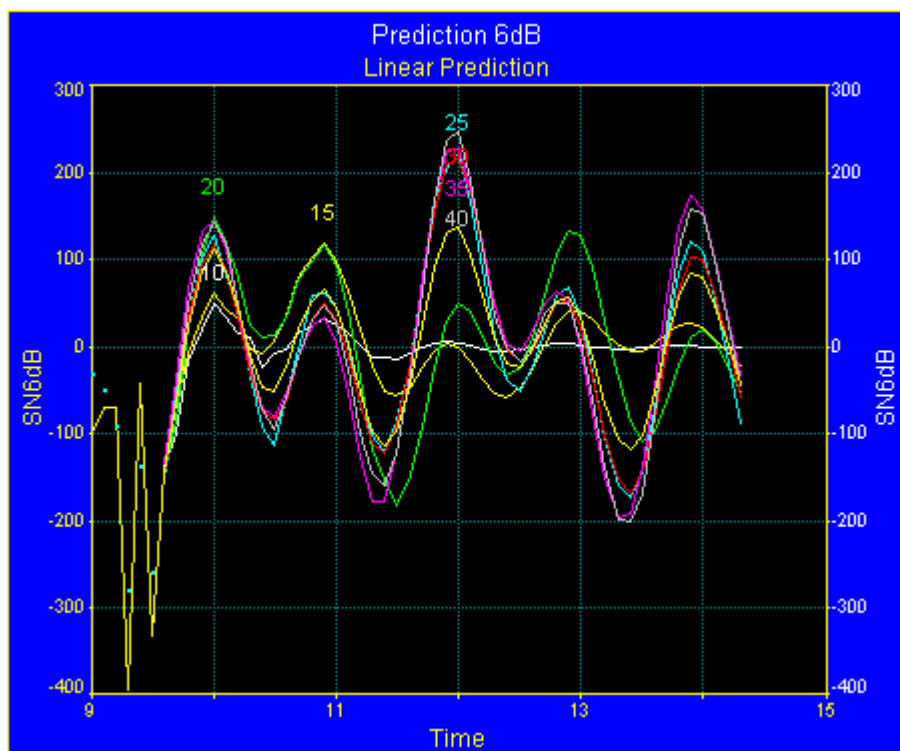


Click OK to update the signal space to 6.



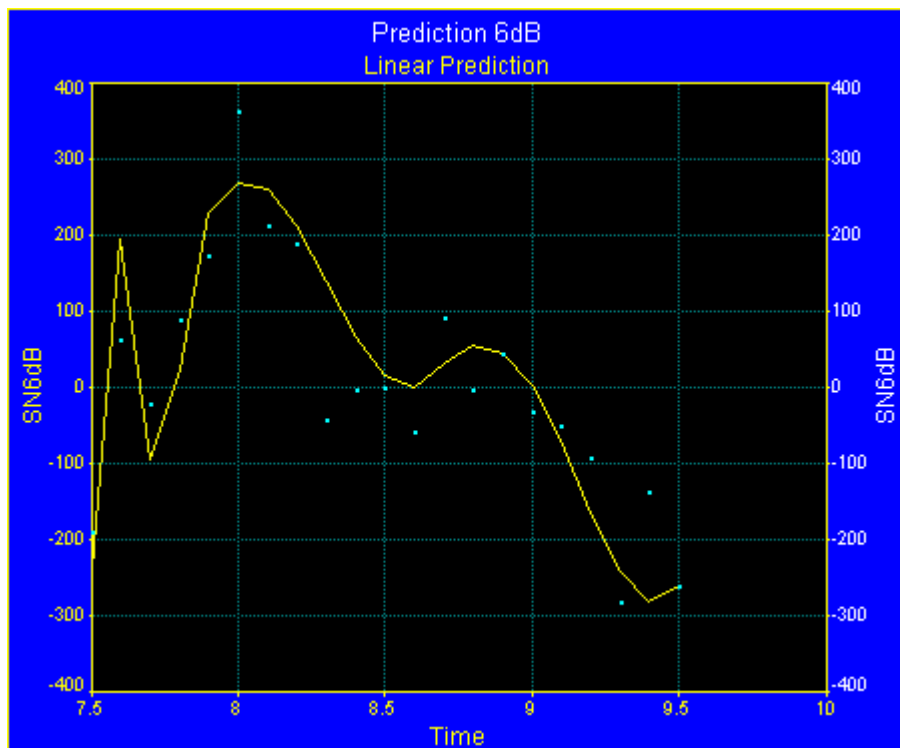
Change the signal space to 8 and then back to 6. Note that there are only minor differences in the prediction. The prediction is certainly smooth, but predictions using the SVD algorithms will usually be smooth. The same questions that were applicable to traditional AR prediction also apply to SVD-based algorithms.

Uncheck Single Order. Enter 10 for min, 40 for max, and 5 for inc. Zoom in on only the predicted portion of the graph.



The model orders 25, 30, 35, and 40 are all very close.

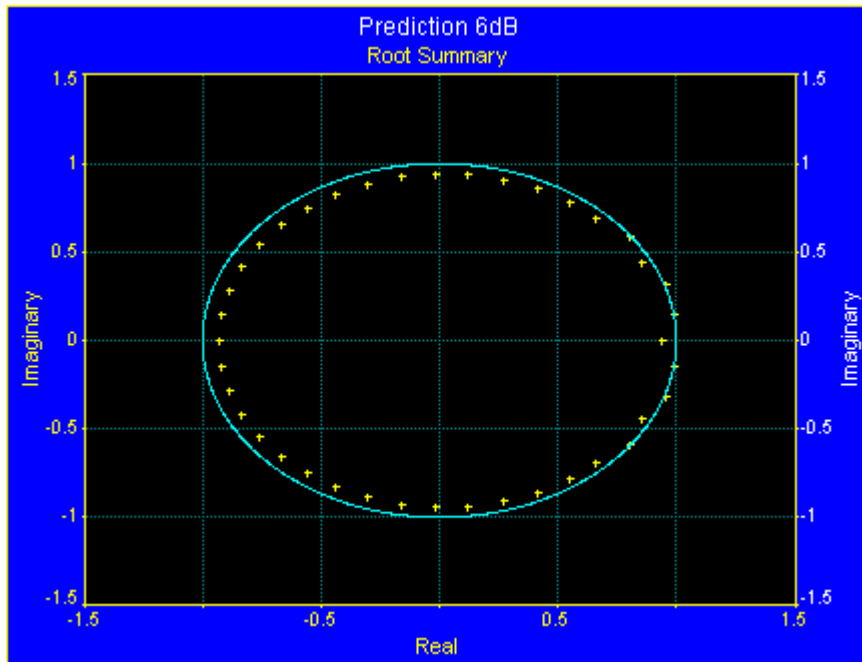
Check Single Order. Enter 40 for the order. Right click in the graph area to restore standard scaling. In the Data Processed box, enter 7.8 for x end and set the Predicted Points n to 17. Zoom in on the predicted region.



The predicted curve looks almost like a smoothed data sequence.

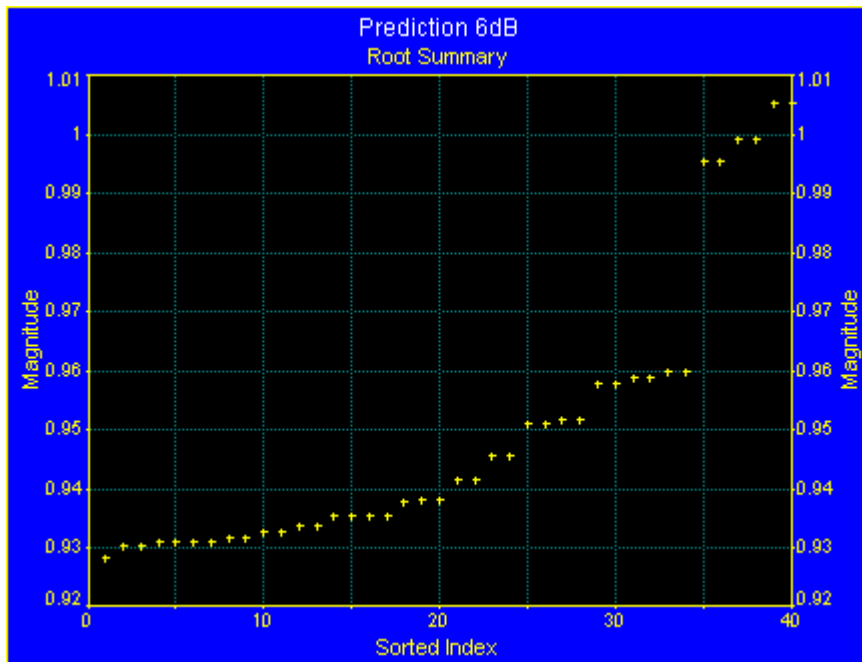


Click the **Plot Roots** button.



Only 2 roots are outside the unit circle now. Six are very near the unit circle and the remainder are in the interior.

Check the **Magnitude** box.

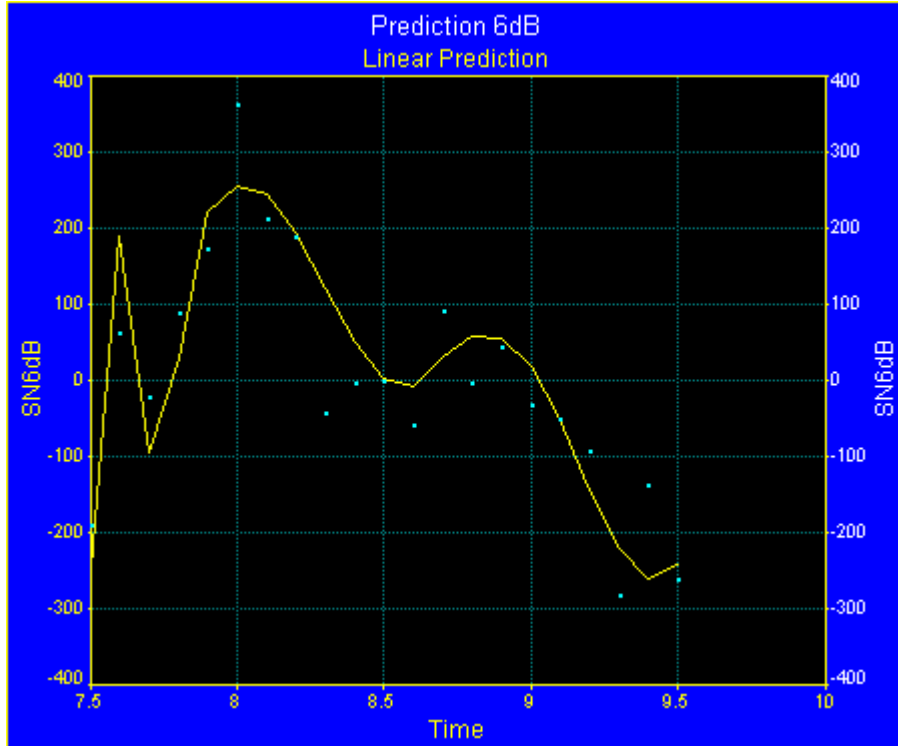


Now we have the three harmonics centered about the unit circle and the roots associated with noise well

toward the interior. Whether the two outlier roots are reflected into interior of the unit circle or mapped to the unit circle proper, they will still be very close to the other two harmonics. Since this profile offers a clear indication these roots are associated with harmonics, we will map these roots to the unit circle.



Click OK to close the Complex Roots plot. For the Stabilization, select Unit Circle Out.

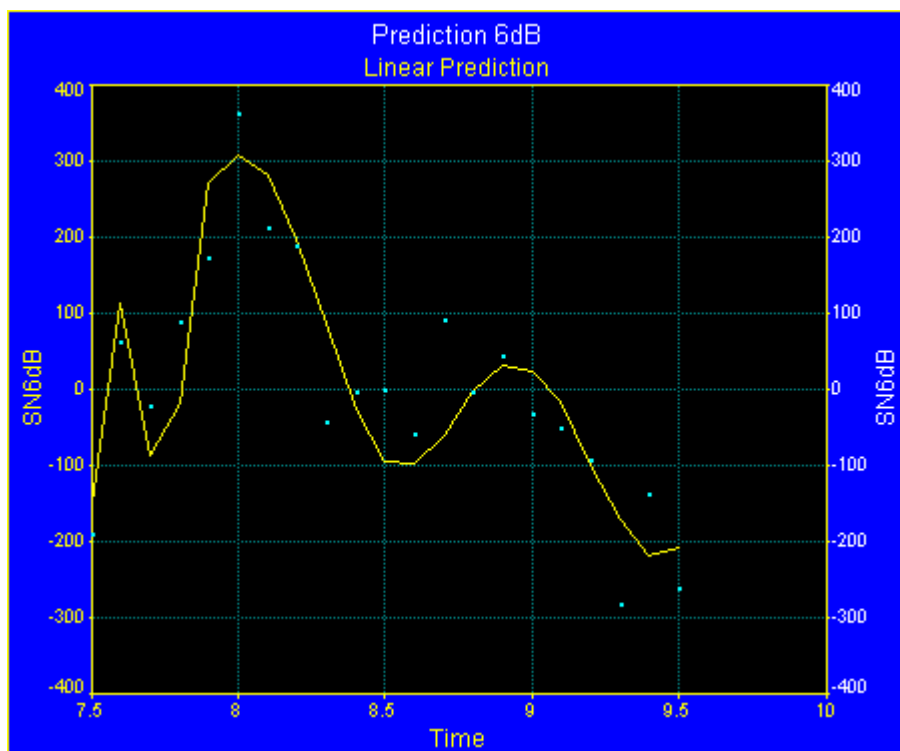


The stabilization produced only minor changes.

Influence of Noise

A prediction model should be stable under the influence of added white (Gaussian) noise.

Set the S/N dB to 6.



This adds another 50% random Gaussian noise to the data on top of the 50% noise originally present. Note that the plotted data remains the original. The AR coefficients, however, are now computed based on the data with this specified amount of added noise. The AR filter is still offering very good predictions, despite this large measure of added noise.

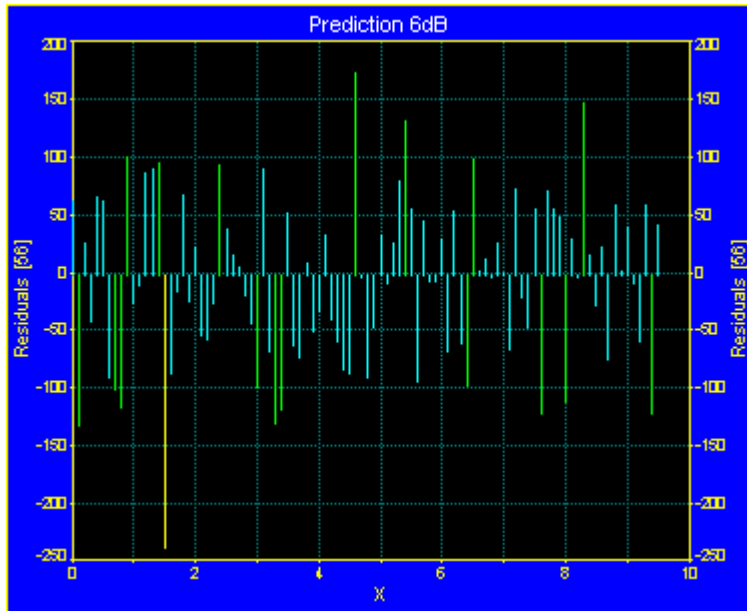
Set the S/N dB back to **300**, the floating point double precision significance limit where no noise is added. Right click in the graph area to restore standard scaling. Set x end to **9.5** to process the full data series.



Click on the **View Residuals** button.



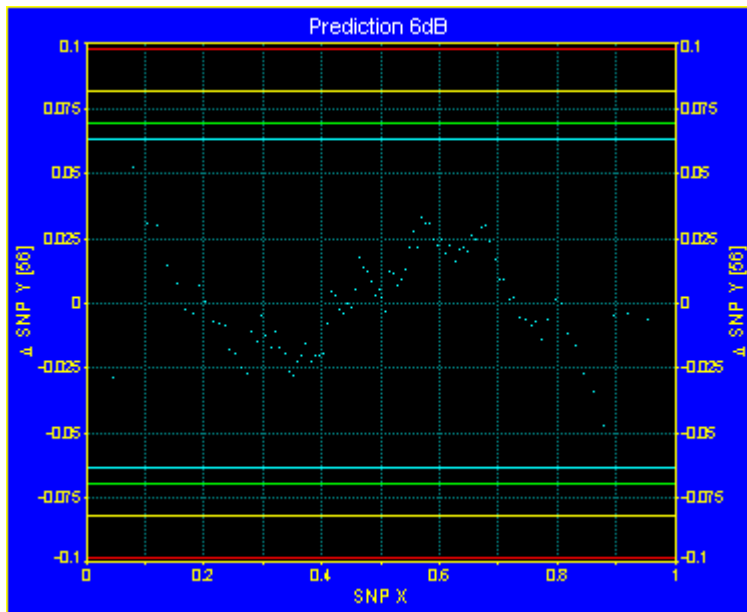
In the Residuals window, click on **Display Basic Residuals**.



The AR model order fit should produce residuals that are comparable to a good parametric fit. That is, there should be no systematic trends in the residuals and they should be normally distributed.



In the Residuals window, click on Display Residuals in Stabilized Normal Probability Plot.



Here the residuals show no apparent systematic trend and are easily judged normally distributed.

Fit Statistics

Fit statistics are also reported. Note that the r^2 is 0.73, very high for a data set with 50% random noise added.

Not all data can be predicted from a linear combination of previous values. Truly stochastic data (all noise

and no deterministic signal) will have very poor AR fit statistics.



To preserve the current settings, click **OK** to close the AR Modeling procedure. Answer **No** to updating the data table.



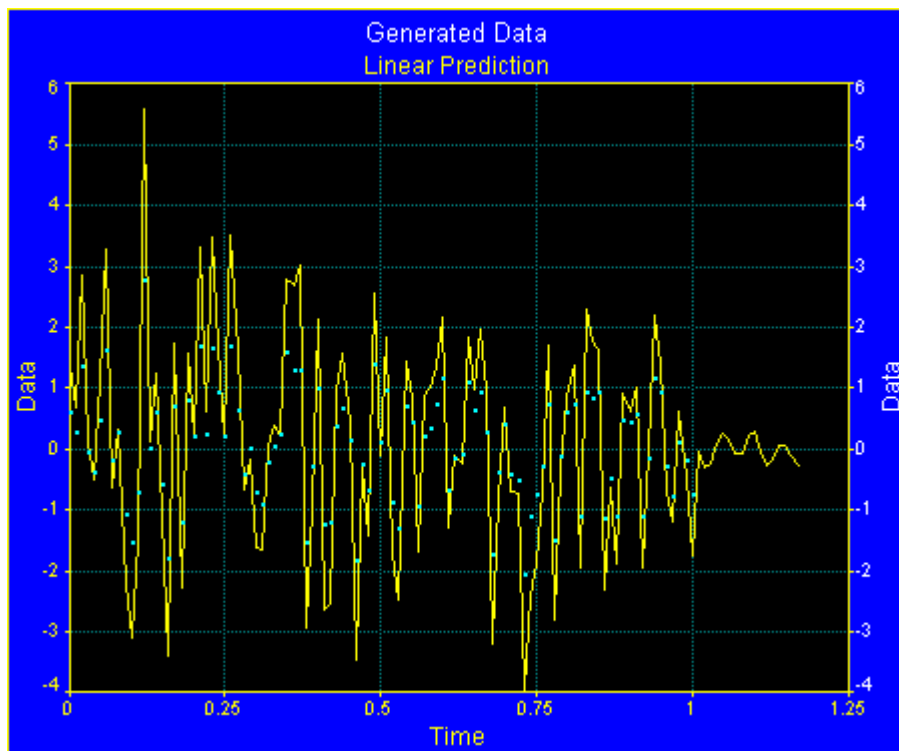
Select the **Generate Data** option in the Edit menu or main toolbar. Enter $Y=IF(X==0,1,0)$ as the data function. Set the Gaussian Noise % to **1000**. Be sure X goes from **0** to **1** with a **0.01** increment. Click **OK**.



Click **OK** to accept the generated data and answer **Yes** to update the data table. Click **OK** to accept the titles.



Return to the **AR Modeling and Prediction** option.



This particular white noise data set has an r^2 value of 0.011. The value you see will be different, but it should be very low. An AR fit is no different from a parametric one. Unless you have strong fit statistics, the model is not valid.



Click on **Cancel** to exit the procedure. Exit the program by closing the main window or by the **Exit** item in the File menu.

Predictions in the Real World

The AR prediction algorithm is useful for processing data containing significant deterministic components. Data that are indistinguishable from noise will not be predicted successfully. Even when deterministic components are present, they may be too complex to be modeled by a linear combination of previous values.

TableCurve 2D's prediction procedure offers robust effective algorithms that can manage a large measure of observation noise. The key to valid predictions using them rests almost entirely on whether a successful fit to the underlying data trend can be realized. Some data sets will be favorably described by an AR model and others will not.