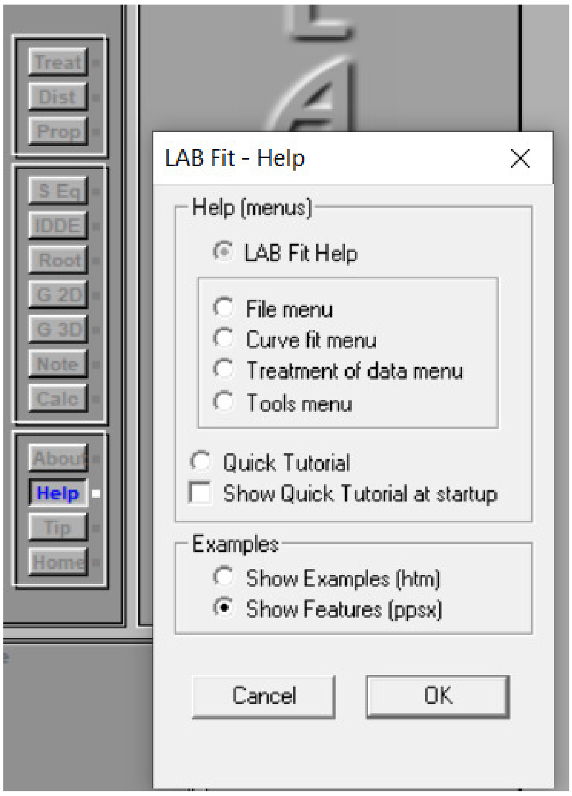
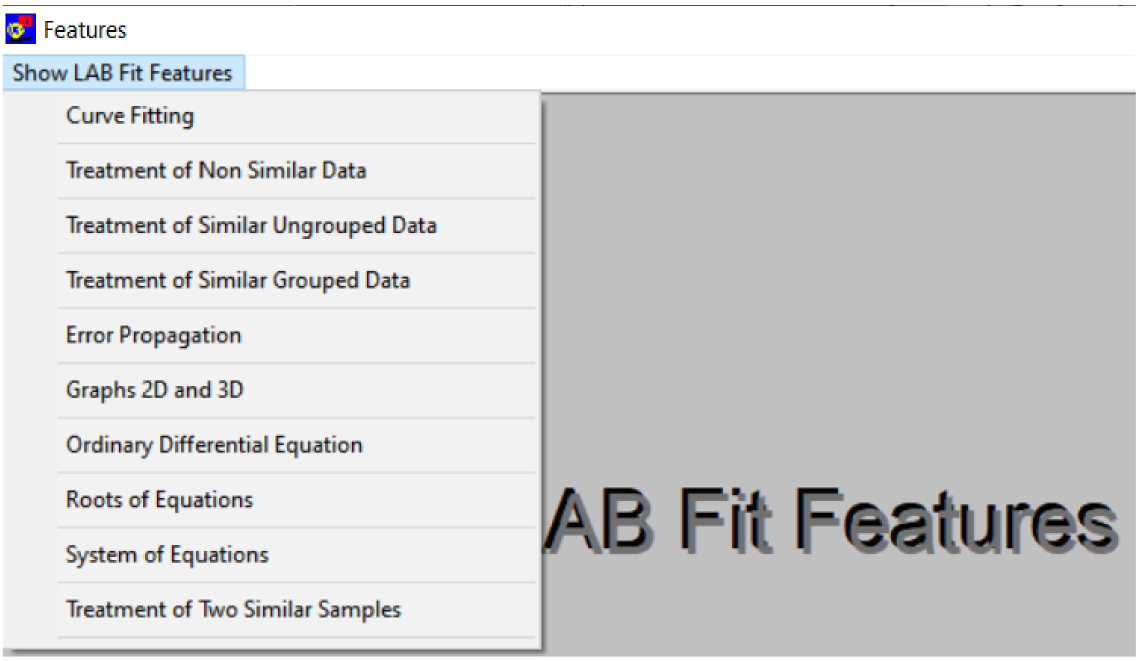
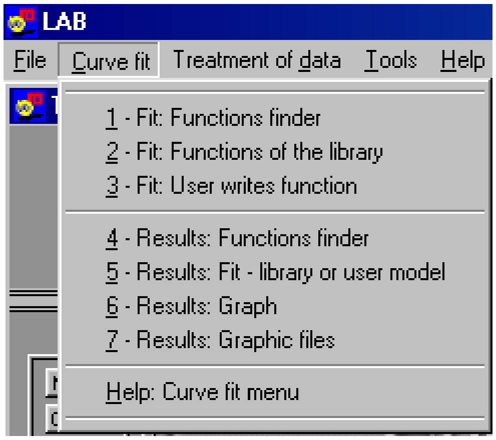
Clicking "Help" and choosing "Show Features (ppsx)"…

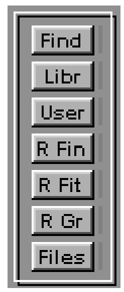


…you will watch several movies with help about…



The "Curve fit" menu





Once the **data set has been informed and is opened, that is, in focus**, the user has three fit options (options number **1**, **2** and **3** from the "Curve fit" menu).

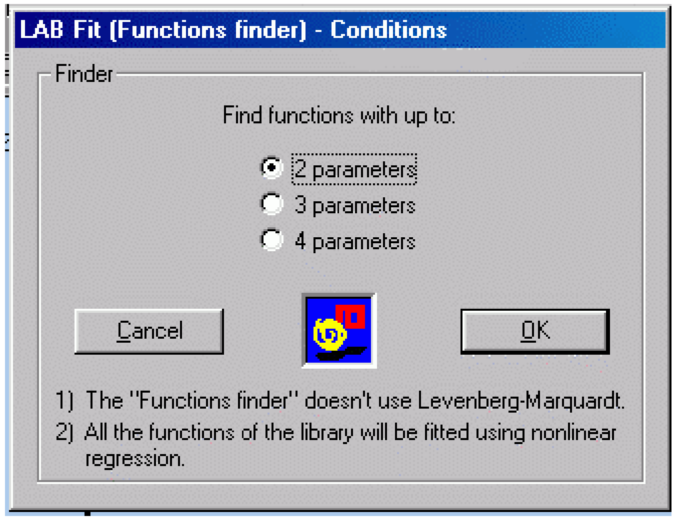
The **first** option is the "**Functions finder**" (corresponding to the button called "Find") that automatically fits all the library functions to the data set, selecting the 10 best results.

The **second** option is the "**Functions of the library**" (corresponding to the button called "Libr") that automatically fits a pre-existing function, defined at the library and chosen by the user, to the data set.

The **third** option is the "**User writes functions**" (corresponding to the button called "User") that makes it possible for the user to write and fit its own function. Let's go now to the details of each of these three options.

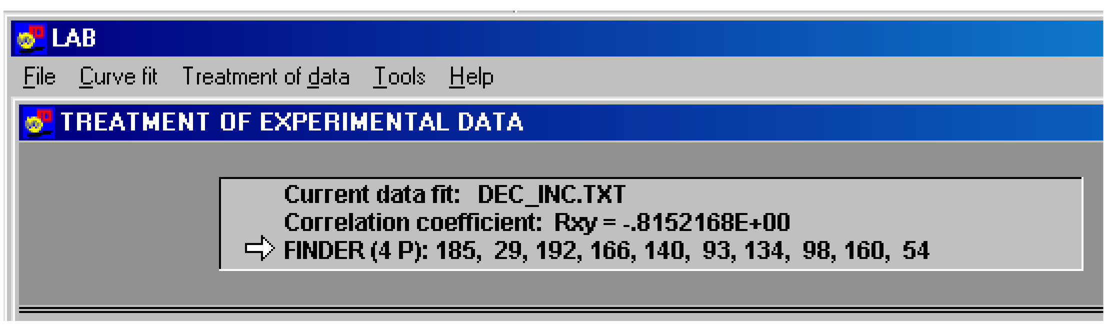
1 - Curve fit: Functions finder (Find)

If you do not know which is the adequate function for your data, the **LAB Fit** almost always finds a good model. To use this option, after informing the data, click on the "Curve fit" menu, and choose the option n° 1, called "Functions finder" (or click on the button called "Find"). Then, the following dialog box will appear:



In this dialog box, the user has the option to select the functions of the library that have only **2** parameters, or **2** and **3** parameters or also **2**, **3** and **4** parameters.

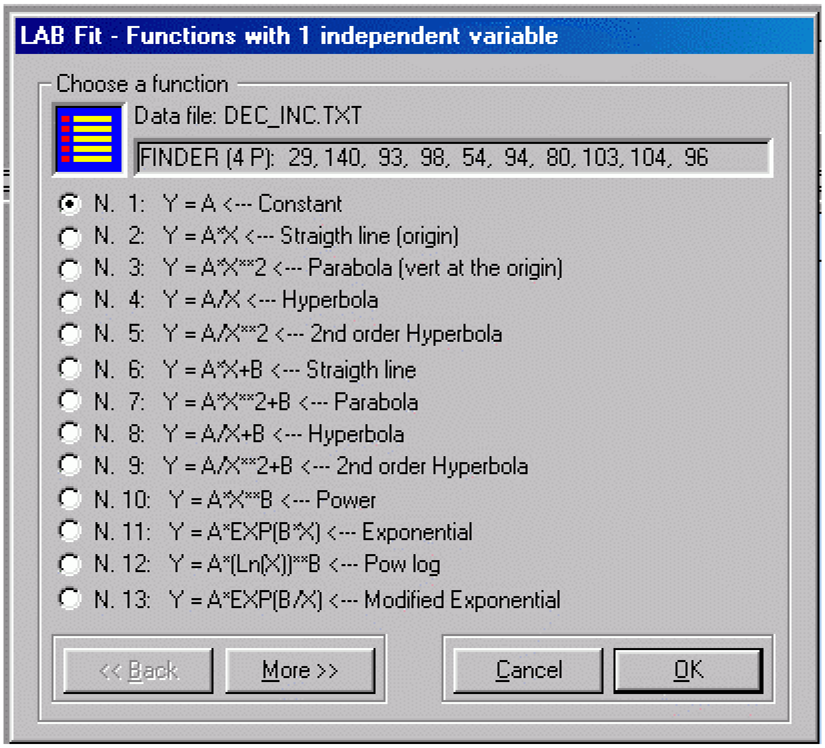
After choosing the quantity of parameters, after clicking on "OK", the **LAB Fit** immediately initiates the test of all the models of the library and gives, at the end, in descending order, the numbers of the 10 best functions. The best results are selected through the comparison of the reduced chi-squared values of the fits. Then, the user can choose a function among the best results and perform the complete fit using the "Functions of the library" option or clicking at the "Libr" button. Once that the "Finder functions" option has been used, the obtained results are added to the data set and they will always be available, as shows the picture at next:



In focus at the main screen of the **LAB Fit** it is possible to see the name of the opened file (current data fit: DEC_INC.TXT) and, for **one single independent variable**, there is also the information about the correlation coefficient between the values of x and y. We can see, also, all the information related to the "Functions finder" option (4 parameters) through the list that shows the number of the 10 best functions for the file DEC_INC.TXT. Such list will only be lost if the data set be modified, using the following options of the "File" menu: "Edit current data file" (corresponding to the "Edit" button) followed by the "Apply the alterations" option (corresponding to the "Appl" button).

2 - Curve fit: Functions of the library (Libr)

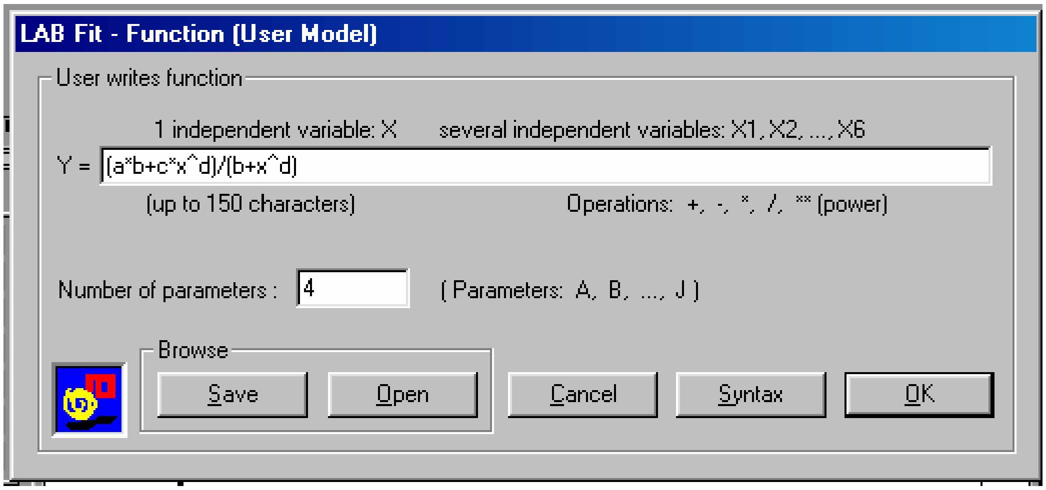
To use this non-linear regression option, the user must choose the "Functions of the library" option at the "Curve Fit" menu or click on the button called "Libr". By doing this, a dialog box containing a miscellaneous list with more than **200 functions with 1 independent variable** and with almost **300 functions with 2 independent variables** becomes available, as shows the dialog box at next:



In this dialog box the data filename appears and, in case the user has already used the "Functions finder" option ("Find" button), the 10 best functions that fit to the data set also appears, as well as the number of the parameters used in the search of these functions (in case of the picture, 4 P). The initial list contains 13 simple functions with one independent variable (the functions complete list is in the final part of this text), and in case the user wants to continue the function selection process, click on the "More" button. Once the function was chosen, click on the "OK" button to initiate the real fit process. The curve fit phase will be described after the next item, at the section called "**The real fit**".

3 - Curve fit: User writes function (User)

To use this non linear regression option, the user must write the second part of the fit function at the first edit box of the dialog box shown at next, obeying the FORTRAN language syntax (that, for arithmetic expressions, is similar to all the other programming languages). Besides writing correctly the second member of the fit function, the user must write the number of the parameters to be determined. See:



At the **first edit box**, it was written, in FORTRAN, the second member of the function that will be fit (the power could also be represented by ^ instead of **). The first unknown parameter is "**A**", the second is "**B**" (always on alphabetical order, starting by **A**) and this function, in particular, has up to the parameter "**D**", what means that the number of parameters is equal to 4 (see this declared at the **second edit box**). In case of functions with 1 independent variable, this variable must be represented by x (if there were 2 independent variables, they would be x1 and x2, and so on).

With these two fields correctly filled, the user can save these information, for future utilizations, clicking on "Save".

Once this information was saved, its future reutilization will be done by clicking on "Open" and selecting the file where such information was stored. The **LAB Fit** has already initiated the build of the user library, with something around **200 functions of 1 independent variable**, **sorted by type**, and something around **270 functions of 2 independent variables**, also **sorted by type**.

To perform the fit, that is, to determine the parameters of the unknown parameters, the user must click on "OK", at the previous dialog box. Then, it will be initiated the relative sequence of the **real fit**.

Fortran Syntax

Operations:

Addition: +

Subtraction: -

Multiplying: *

Division: /

Power: ** or ^

Functions applied to a value x

Sine of x: **sin(x)**

Cosine of x: **cos(x)**

Tangent of x: **tan(x)**

Arc which the sine is x: **asin(x)**

Arc which the cosine is x: **acos(x)**

Arc which the tangent is x: **atan(x)**

Hyperbolical Sine of x: **sinh(x)**

Hyperbolical Cosine of x: **cosh(x)**

Hyperbolical Tangent of x: **tanh(x)**

Sine of x (x in degrees): **sind(x)**

Cosine of x (x in degrees): **cosd(x)**

Tangent of x (x in degrees): **tand(x)**

Natural Logarithm of x: **log(x)**

Logarithm of x at the base 10: **log10(x)**

Exponential of x (e powers x): **exp(x)**

Root square of x: **sqrt(x)**

Absolute value of x: **abs(x)**

The real fit

The **LAB Fit** makes it possible the fitting of functions with one and with several independent variables (up to 6) with or without uncertainties at the variables (dependent and independent).

Points with unknown uncertainties: if there aren't uncertainties neither in y nor in x, the uncertainties of x will be automatically considered equal to zero, and the y ones will initially be set 1. After a pre-fit, a second one will be executed, imposing the value 1 for the reduced chi-squared, what makes the fit credible and makes it possible to calculate a common uncertainty to all the values of y.

Points with uncertainties only in y: in case there are uncertainties in y and there aren't in x, the fit will be, then, performed for this situation.

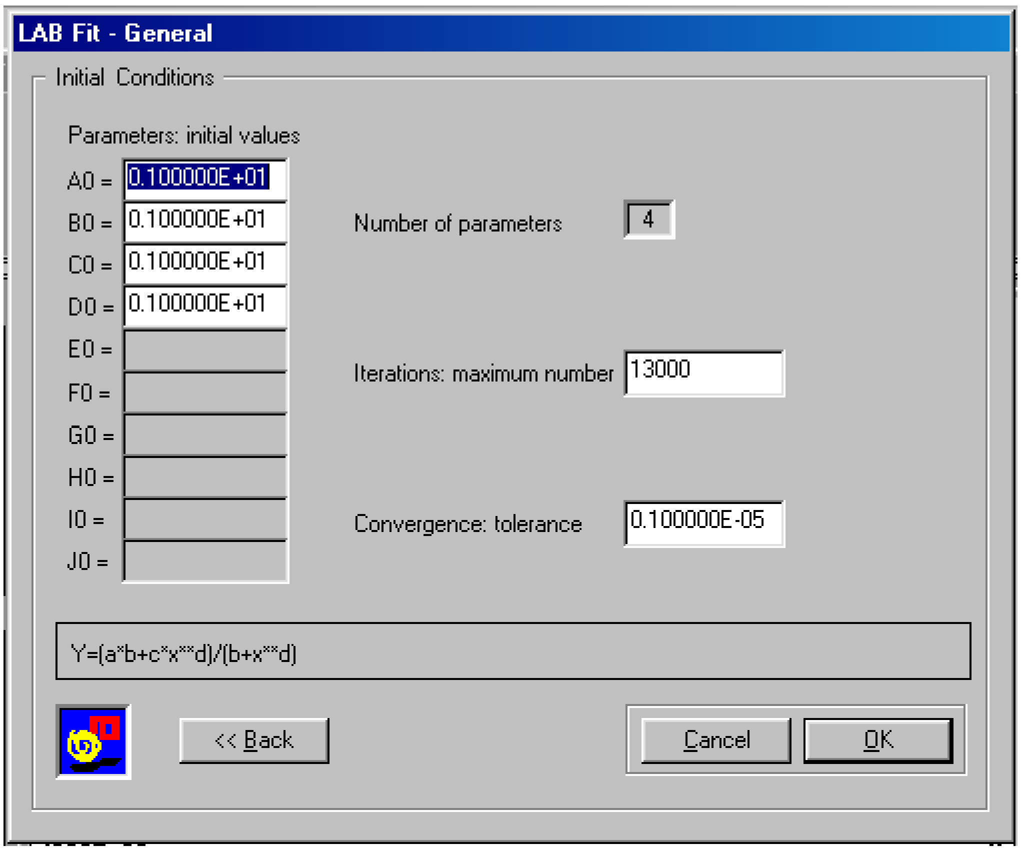
Points with uncertainties only in x: if there are uncertainties in x but not in y, a pre-fit will be performed, attributing a residual value for the uncertainties of y (equal to 0.0000001). After, a new fit with the uncertainties transferred from x to y by error propagation will be executed, using first order approximation.

Points with uncertainties in x and in y: if there are uncertainties in x and in y, a pre-fit considering the uncertainties only in y will be executed. After, a new fit with the uncertainties transferred from x to y by error propagation will be executed, using first order approximation.

In all the cases previously seen, the **statistical weights** are defined as the inverse of the square of the final uncertainties in y.

As it was previously seen, if there aren't uncertainties neither in x nor in y, the **LAB Fit** imposes a new condition of the reduced chi-squared equal to 1 for the fit. Otherwise, in the cases in which y and/or x have uncertainties, the user will have the option to impose or not a chi-squared equal to 1.

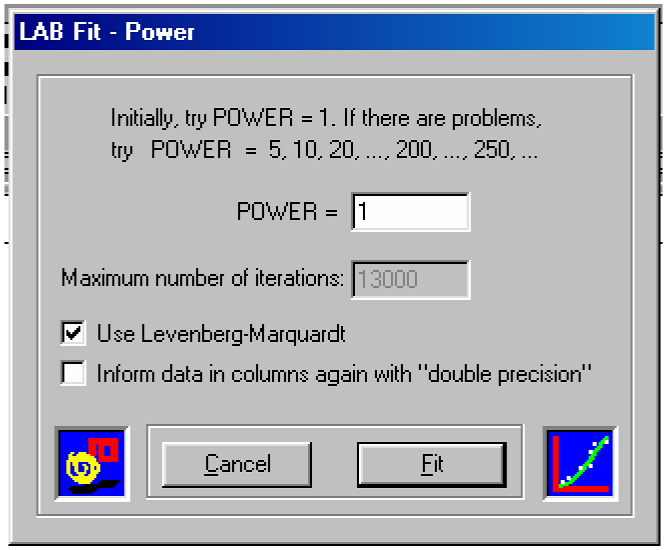
Once defined the fit function, through the "Functions of the library" option ("Libr" button) or through the "User writes functions" option ("User" button), the following dialog box will appear aiming to establish the initial parameters values and the general conditions for the non linear regression (maximum number of iterations and tolerance for the convergence):



If the user makes a reasonable estimate for the initial values **A0**, **B0**, etc, the values stipulated by the "User writes functions" option (equal to 1) or the values calculated by the "Functions of the library" option must be erased, and the user must write its **own estimate**. In case any reasonable estimate is not known, the user must try the non-linear regression with the initial values defined by the **LAB Fit** itself.

As the process is iterative, to avoid something similar to a loop, **the maximum number of iterations** must be stipulated (the default is 300, but it is possible to increase this number, as it is shown at the previous dialog box). The user can change also the **tolerance for the convergence** for the iterative process (the default is 1.0E-06).

With everything ok at the previous dialog box, click on “OK”. It will be starting now **the real fit** phase, and the following dialog box will appear:



“Power” Parameter. If the initial values of the parameters are adequate, the user must click on the “Fit” button so the fit can be executed. In case the user doesn’t make a clear idea if the initial values are good or not, the “power” parameter must be increased (it can be increased up to the limit defined by the maximum number of iterations, set by the user, and shown right below the “power” value). The increase of the “power” parameter increases also the probability of the iterative fit convergence, the process becomes slower however.

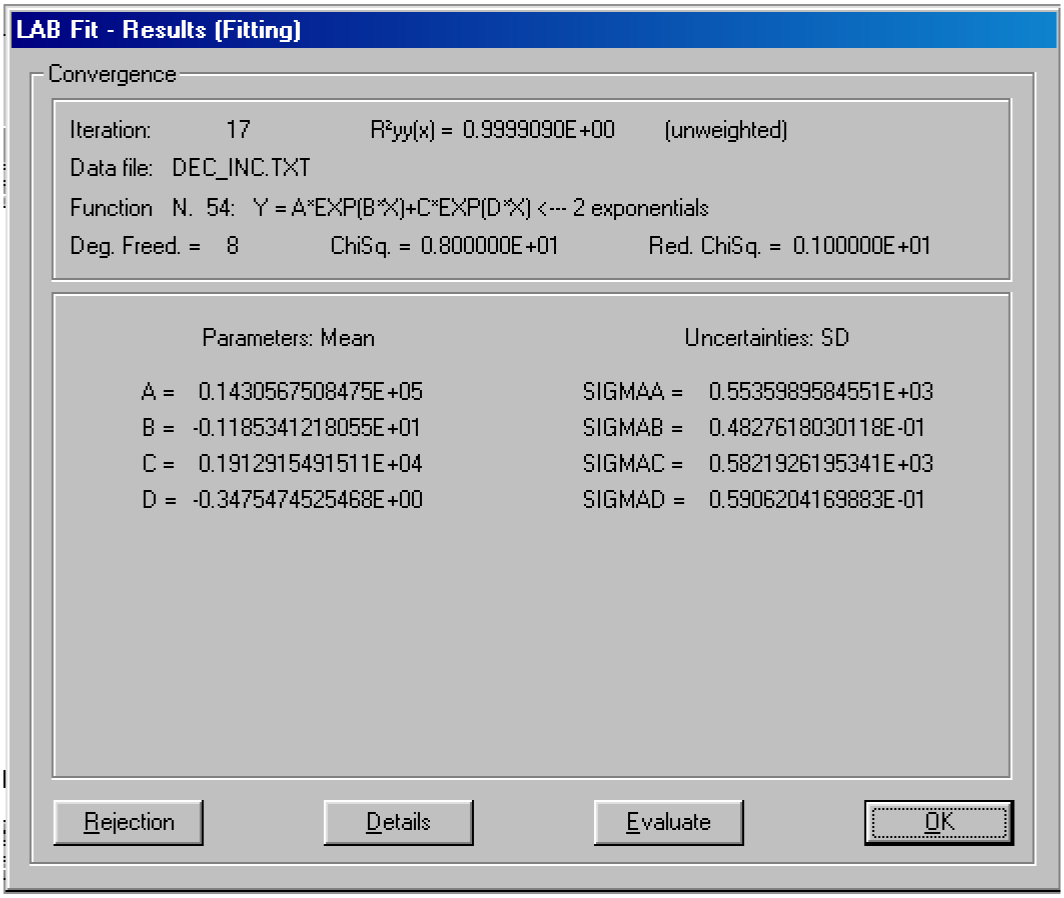
Levenberg-Marquardt. It is a well known algorithm that increases the possibility to have convergence, by lowering the possibility of overflow and other mishaps, predictable in this type of fit. So, this check box must be left untouched and checked (its default state), until the user acquires more confidence on the use of the **LAB Fit**. After, if it is desired, the user can try the fit without using the Levenberg-Marquard algorithm.

Attention: a common error done by beginners is to think that the fits will always finish successfully. It’s not quite that way. For the fit of a function to experimental data be successful, **two demands** are needed.

The **first demand** is related to the function chosen by the user. Such function must be really compatible with the experimental points.

The **second demand** is related to the initial values that cannot be so bad that will not make possible for the **LAB Fit** to search for a convergence.

The fit will be performed according to the specifications of the data set and, at the end, if everything ends up to be ok, a dialog box with the fit results **summary** will appear, similar to the dialog box shown at next:



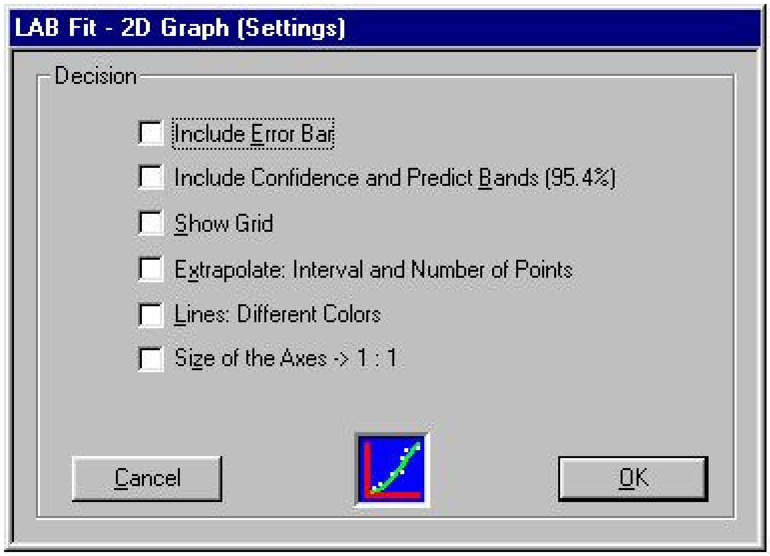
In this dialog box it is possible to see the **number of iterations** for the convergence of the iterative process (in this case, 35). It is possible to see, also, the **correlation coefficient** between the Yi data and the corresponding values obtained through the fit function Y(Xi), symbolized by $R^2yy(x)$. Besides, it is also informed the **name** of the data file and also the expression of the fit **function**. The number of **degrees of freedom** of the fit and also the **chi-squared** is informed. Beyond this information, it is also presented the list with the values of the **fit parameters** and their respective **uncertainties**.

The fit detailed results, including the covariances matrix, can be obtained through the "Results: Fit – library or user model" option or through the "R Fit" button.

When "OK" is clicked at the previous dialog box, the graph of the fit function will be drawn, as will be shown at next, for both 2D and 3D cases. Initially the options to draw the graph of a function fit in 2D will be shown and, in sequence, the same considerations will be done for 3D.

2D Graph of the fit function

There are several options to draw a 2D graph of a fit function, as it is shown at the dialog box at next:



This dialog box makes it possible for the user to define how the graph will be drawn.

The **first** check box, if checked (true) will include an **error bar** into each experimental point. For points with unknown uncertainties, these error bars are calculated from the fluctuations of the experimental points around the fit function, imposing a reduced chi-squared equal to 1 for the fit. So, as better the fit is, smaller will be the error bars.

For points in which only the uncertainties of y were informed, the error bars represent these uncertainties.

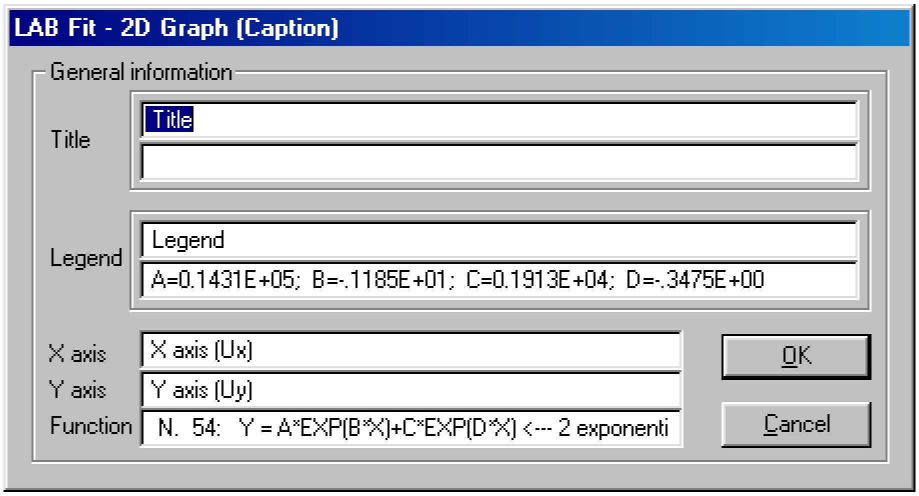
If the uncertainties in x and in y are informed, then, through uncertainties transfer (error propagation), the final uncertainties in y will be calculated, and these final uncertainties will be represented.

The **second** check box makes it possible for the user to include at the graph an **uncertainty band** (confidence band) with 95% of confidence. This uncertainty band consists basically in two auxiliary lines, limiting above and below the main line related to the fit function. These lines define the region where y(x) must occur, with 95% of confidence.

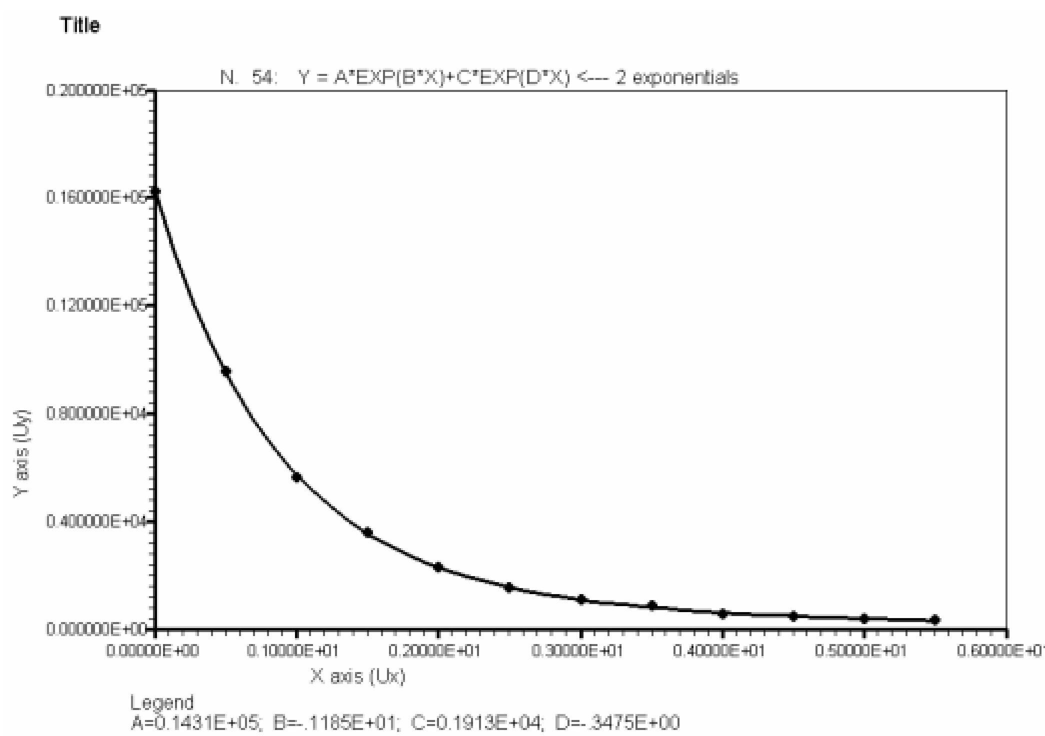
The **third** check box makes it possible to trace a **grid** at the XY plane of the graph, whereas the **fourth** one makes it possible to stipulate the values of X to trace the graph using **extrapolation** of the fit function.

In case of the inclusion of the uncertainty band (confidence band) the **fifth** check box makes it possible to trace a central line with a **different color** from the lines that define the band.

Once established the conditions at the previous dialog box, when "OK" is clicked, the 2D graph will be drawn and, simultaneously, a new box for the final characterization will appear (title, legend, axis and fit function definition):



After the graph characterization, it can be printed or **pasted into a document** like the picture below, that has its original reduced in 20%.

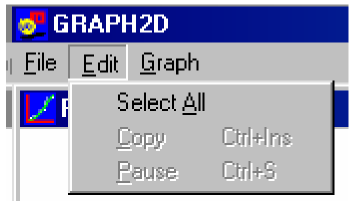


To print the 2D graph, there are two options.

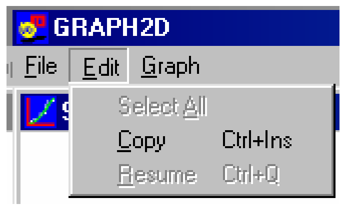
To use the **first option**, "Print window", at the "File" menu, must be clicked, but this option only works to print a **sketch**, because the printing is not of good quality: the printed picture will have the same resolution shown on screen.

The **second option** makes it possible to print a graph with **a better quality than the previous one**, what can be done by clicking on the "Superzoom in" item of the "Graph" menu. The graph will be, then, drawn again, but with a much higher resolution than the original one, occupying a space much bigger than the available screen size. After that, at the "File" menu, the user must click on "Print graph (superzoom)".

To **paste** the graph at a text file the user must choose, at the "Edit" menu, the "Select All" option as it is shown at the picture at next:



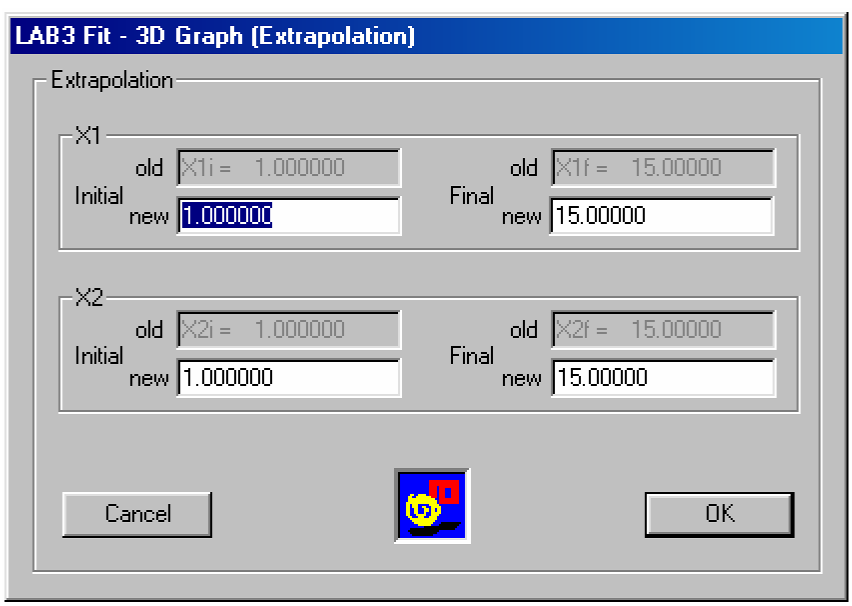
After, at the same menu "Edit", "Copy" must be clicked, as it is shown at next.



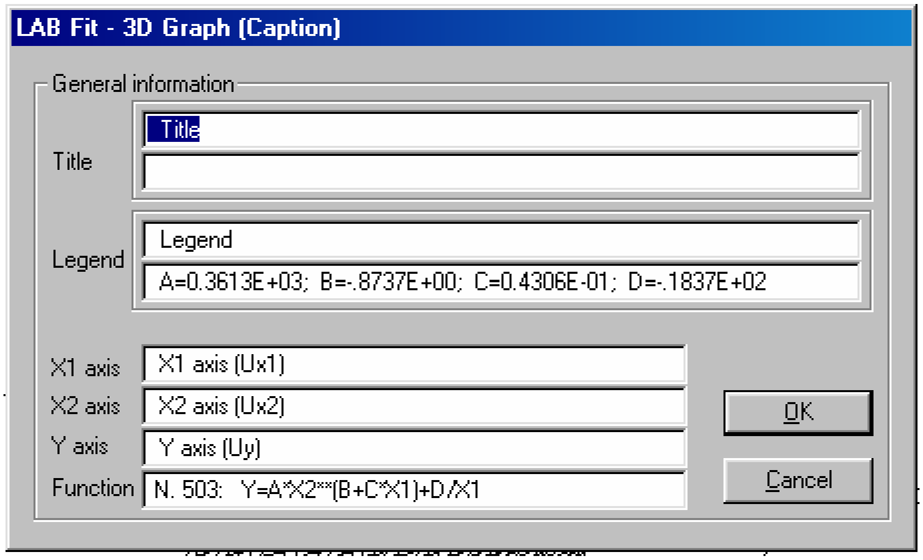
Then, it is just a matter of opening the text file, and pasting the image that was stored at the clipboard.

3D Graph of the fit function

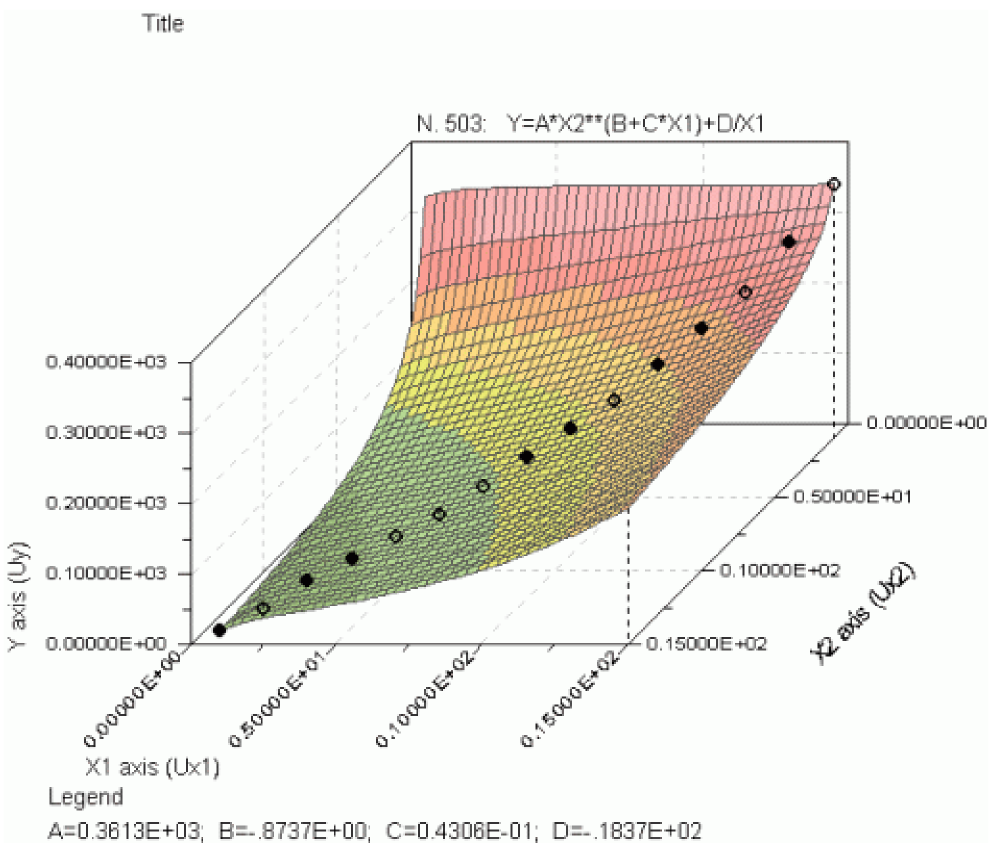
As soon as the real fit is finished, the graph program is automatically called. In case the fit function has 2 independent variables, the following dialog box makes it possible for the user to draw the graph **with or without extrapolation**:



The 2 independent variables are X1 and X2. The "old" values are the original ones from the data set, however the user can modify the "new" values, in case if it is desired, to draw the graph with extrapolation of the fit function. With the "new" values of the two independent variables already informed; click on the "OK" button to obtain the 3D graph. The graph will be drawn and it will appear, then, a new dialog box for the final characterization (title, legend, axis and fit function definition). This box can be seen in the picture at next:



An example of a function graph fit by the **LAB Fit**, with two independent variables, can be seen at next, at a picture where the original size was reduced:



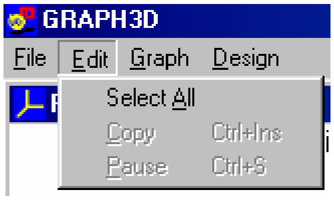
Attention: points represented by filled circles are above (or on) the surface and not-filled circles are below the surface.

To print the 3D graph, there are two options.

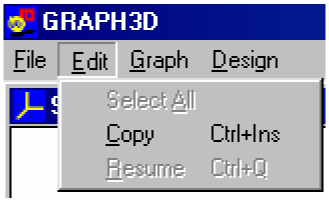
To use the **first option**, "Print window", at the "File" menu, must be clicked, but this option only works to print a **sketch**, because the printing is not of good quality: the printed picture will have the same resolution shown on screen.

The **second option** makes it possible to print a graph with **a better quality than the previous one**, what can be done by clicking on the "Superzoom in" item of the "Graph" menu. The graph will be, then, drawn again, but with a much higher resolution than the original one, occupying a space much bigger than the available screen size. After that, at the "File" menu, the user must click on "Print graph (superzoom)".

To **paste** the graph at a text file the user must choose, at the "Edit" menu, the "Select All" option as it is shown at the picture at next:



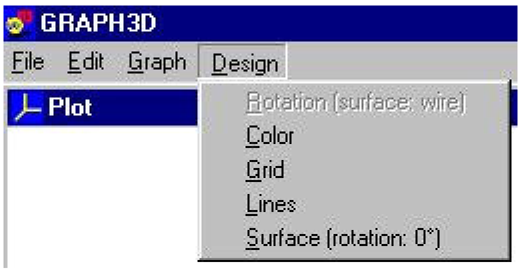
After, at the same "Edit" menu, "Copy" must be clicked, as it is shown at next.



Then, it is just a matter of opening the text file, and pasting the image that was stored at the clipboard.

The "Design" menu of Graph3D

In case the user is doing a 3D graph, several options can be chosen, that are shown at the "Design" menu, at next.



The “Rotation (surface: wire)” option makes it possible to rotate the graph in up to 45 degrees (the default is 0 degree) but does not work for a solid mesh which is the default option. Whereas the “Color” option makes it possible to choose among six distinct colors (standard, blue, red, magenta, green and gray, the default is standard) for the drawing of the **wire frame** or the **solid mesh** that indicates the surface. The “Grid” option makes it possible to draw grids at the X1-Y, X2-Y and X1-X2 planes (the default is with grid) whereas that, clicking on “Lines”, the user can indicate with how many lines the surface must be drawn (from 10 up to 100, the default is 40).

4, 5, 6, e 7 - Curve fit: Results

The results for a fit already done, but with the data still in focus (opened), are obtained through the options number **4**, **5**, **6** and **7** from the “Curve fit” menu. The access to the results and their comments will be described at next.

Finder

When executing the “Functions finder” option (corresponding to the button called “Find”) for a determined data set, the user can access the results of all the fits, while the data set is opened, without the necessity of a re-execution. This is done through the “Results: Functions finder” option (corresponding to the button called “R Fin”) that opens the file containing the information of all the fit functions. Even lately, in a new execution, the 10 best results will be available because this information is added to the data set. Such information will be available on screen, as it is shown at the previous “**1 – Curve fit: Functions finder (Find)”** item. There, in projection at the **LAB Fit** main screen, it is possible to see the file in focus (current data file) and, for a single independent variable, there is also the information about the correlation coefficient between the values x and y. It is possible to see, also, the information related to the “Functions finder” option (4 parameters). Such information is kept at the list showing the number of the 10 best results for the file DEC_INC.TXT. The list will only be erased if the data set be modified by the user, through the use of the “Edit current data file” option (“Edit”), from the “File” menu, followed by the “Apply the alterations” option (“Appl”).

Curve fit

When executing the “Functions of the library” (“Libr” button) or “User writes function” (“User” button) options, the fit results can be accessed through the “Results: Fit – Library or user models” option (button called “R Fit”). There are two access options for the results of the fit:

summary of the results (mean value and parameter uncertainties, chi-squared, and the correlation coefficient between y and y(x)), or even

detailed results that include various information within a file, including the covariance matrix between the fit parameters.

Beyond these results, the “Results: Graph” (button called “R Gr”) option makes it possible to redo the graph of the fit function while the data and fit function are in focus.

To finish, during the fit, several files needed to draw the graph are generated, and all of them are available through the “Results: Graphic files” option (button called “Files”).

LIBRARY FUNCTIONS

WITH 1 INDEPENDENT VARIABLE

N. 1: Y = A <--- Constant
N. 2: Y = A*X <--- Straight line (origin)
N. 3: Y = A*X**2 <--- Parabola (vertex at the origin)
N. 4: Y = A/X <--- Hyperbola
N. 5: Y = A/X**2 <--- 2nd order Hyperbola
N. 6: Y = A*X+B <--- Straight line
N. 7: Y = A*X**2+B <--- Parabola
N. 8: Y = A/X+B <--- Hiperbola
N. 9: Y = A/X**2+B <--- 2nd order Hyperbola
N 10: Y = A*X**B <--- Power
N. 11: Y = A*EXP(B*X) <--- Exponential
N. 12: Y = A*(Ln(X))**B <--- Pow log
N. 13: Y = A*EXP(B/X) <--- Modified Exponential
N. 14: Y = A*X**3 <--- Parabola (vertex at the origin)
N. 15: Y = A*X**4 <--- Parabola (vertex at the origin)
N. 16: Y = 1/(A*X+B) <--- Inverse of straight line
N. 17: Y = X/(A+B*X) <--- Inverse of hyperbola
N. 18: Y = A*B**X <--- Modified power
N. 19: Y = A*B**(1/X) <--- Root (mod exponential)
N. 20: Y = A*X**(B*X) <--- Super geometric
N. 21: Y = A*X**(B/X) <--- Modified geometric
N. 22: Y = A+B*LnX <--- Logarithmic
N. 23: Y = 1/(A+B*LnX) <--- Inverse of logarithmic
N. 24: Y = A*X+B*X**2 <--- Parabola (origin)
N. 25: Y = SQRT(A-(X-B)**2) <--- Arc of circle
N. 26: Y = X/(A+B*X**2) <--- Straight / parabola
N. 27: Y = A*(B**X)*X**C <--- Hoerl
N. 28: Y = A*(B**(1/X))*X**C <--- Mod Hoerl
N. 29: Y = A*EXP(((X-B)**2)/C) <--- Gaussian
N. 30: Y = A*EXP(((LnX-B)**2)/C) <--- Log Gaussian
N. 31: Y = 1/(A*(X+B)**2)+C) <--- Cauchy
N. 32: Y = A*(X**B)*(1-X)**C <--- Beta
N. 33: Y = A+B/X+C/X**2 <--- 2nd order Hyperbola
N. 34: Y = A*X+B*X**2+C*X**3 <--- 3nd order polynomial
N. 35: Y = A+B*X+C/X <--- Straight line and inverse
N. 36: Y = A+B*X+C*X**2 <--- Parabola
N. 37: Y = A*((X/B)**C)*EXP(X/B) <--- Gamma
N. 38: Y = 1/(A+B*X+C/X) <--- Inv: straight line and hiperbola
N. 39: Y = SQRT(A-B*(X-C)**2) <--- Ellipse
N. 40: Y = A+B*X+C/X**2 <--- Straight line + hyperbola
N. 41: Y = A+B*X**2+C/X <--- Parabola + hiperbola
N. 42: Y = A*X**(B+C*X) <--- Pow (2)
N. 43: Y = A*X**(B+C/X) <--- Pow (3)
N. 44: Y = A*X**(B+C*LnX) <--- Pow (4)
N. 45: Y = A*X**(B*X+C*X**2) <--- Pow (5)
N. 46: Y = A*EXP(B*X+C*X**0.5) <--- Exp (2)
N. 47: Y = A*EXP(B/X+C*X) <--- Exp (3)
N. 48: Y = (A+X)/(B+C*X) <--- Straight line / straight
N. 49: Y = (A+X)/(B+C*X**2) <--- Straight line / parabola
N. 50: Y = A+B*X+C*X**2+D*X**3

N. 51: Y = A+B*X+C*X**2+D*X**3+E*X**4
N. 52: Y = A+B*X+C*X**2+D*X**3+E*X**4+F*X**5
N. 53: Y = A+B*X+C*X**2+D*X**3+E*X**4+F*X**5+G*X**6
N. 54: Y = A*EXP(B*X)+C*EXP(D*X) <--- 2 exponentials
N. 55: Y = A*(EXP(B*X)-EXP(C*X)) <--- Assoc exp
N. 56: Y = A*X**B+C*X**D <--- 2 pow
N. 57: Y = A+B*EXP(C*X) <--- 1 exp + c
N. 58: Y = A*(1-EXP(B*X)) <--- Assoc exp
N. 59: Y = A*COS(B*X) <--- Cossine
N. 60: Y = A*COS(B*X+C) <--- Cossine
N. 61: Y = A*SEN(B*X) <--- Sine
N. 62: Y = A*SEN(B*X+C) <--- Sine
N. 63: Y = A*TAN(B*X) <--- Tangent
N. 64: Y = A*TAN(B*X+C) <--- Tangent
N. 65: Y = A*ACOS(B*X)+C <--- Inv cossine
N. 66: Y = A*ATAN(B*X)+C <--- Inv tangent
N. 67: Y = A*ASEN(B*X)+C <--- Inv sine
N. 68: Y = A*TAN(B*X+C)+D <--- Tangent
N. 69: Y = A*SEN(B*X+C)+D <--- Sine
N. 70: Y = A*COS(B*X+C)+D <--- Cossine
N. 71: Y = A*COS(D*X)+B*COS(2*D*X)+C*COS(3*D*X)
N. 72: Y = A*COS(1*E*X+D)+...+C*COS(3*E*X+D)
N. 73: Y = A*COS(1*E*X+D)+...+C*COS(3*E*X+D)+F
N. 74: Y = A*X**B+C <--- 1 pow + c
N. 75: Y = A*X**B+C*X**D+E <--- 2 pow + c
N. 77: Y = A*X**B+C*EXP(D*X) <--- Pow + exp
N. 78: Y = A*X**B+C*EXP(D*X)+E <--- Pow + exp + c
N. 79: Y = (A*X**B)*(C-X)**D <--- Mod Beta
N. 80: Y = A*EXP(B*EXP(C*X)) <--- Gompertz
N. 81: Y = A/(1+B*EXP(C*X)) <--- Logistic
N. 82: Y = A/(B+X**C) <--- Langmuir
N. 83: Y = A/(1+B*X**C)**2 <-- Mod Langmuir
N. 84: Y = (A+B*X**C)/(D+X**C) <--- MMF
N. 85: Y = (A+B*X)**C <--- Bleasdale (Shifted power)
N. 86: Y = (A+B*SQRT(X))**C <--- Mod Bleasdale
N. 87: Y = 1/(A+B*X**C) <--- Harris
N. 88: Y = 1/(A+B*EXP(C*X))**D <--- Richards
N. 89: Y = A*EXP(B/X)+C <--- Mod exp + c
N. 90: Y = 1/(A*X+B)+C <--- Inv straight line + c
N. 91: Y = X/(A+B*X)+C <--- Inv hyperbola + c
N. 92: Y = (A*X**B)*(C-X)**D+E <--- Mod Beta + c
N. 93: Y = A*EXP(B*EXP(C*X))+D <--- Gompertz + c
N. 94: Y = A/(1+B*EXP(C*X))+D <--- Logistic + c
N. 95: Y = EXP(A+B/X+C*Ln(X)) <--- Vapor Pressure (mod Hoerl)
N. 96: Y = A/(1+B*X**C)**2+D <--- Mod Langmuir + c
N. 97: Y = (A+B*X**C)/(D+X**C)+E <--- MMF + c
N. 98: Y = (A+B*X)**C+D <--- Bleasdale (Shifted power) + c
N. 99: Y = (A+B*SQRT(X))**C+D <--- Mod Bleasdale +c
N. 100: Y = 1/(A+B*X**C)+D <--- Harris + c
N. 101: Y = 1/(A+B*EXP(C*X))**D+E <--- Richards + c
N. 102: Y = A*EXP(B*X**C) <--- Exp of pow
N. 103: Y = A*EXP(B*X**C)+D <--- Weibull
N. 104: Y = (A+B*X)/(1+C*X+D*X**2) <--- Rational
N. 105: Y = A*X**(B*X**C) <--- Freundich
N. 106: Y = X/(A+B*X+C*SQRT(X)) <--- Gunary
N. 107: Y = 1/(A+B*X**2)**C <--- Inv pow

N. 108: Y = A*(Ln(X))**B+C <--- Pow of log + c
N. 109: Y = A*Ln(X+B) <--- Mod Log
N. 110: Y = A*Ln(X+B)+C <--- Mod Log + c
N. 111: Y = A*X**B+C*EXP(D/X) <--- Pow + mod exp
N. 112: Y = A/X+B*X**C <--- Hyp + pow
N. 113: Y = A/X+B*EXP(C/X) <--- Hyp + mod exp
N. 114: Y = A/X+B*EXP(C*X) <--- Hyp. + exp
N. 115: Y = A*X*EXP(B*X) <-- Exp x straight line
N. 116: Y = A*EXP(B*X)/X <-- Hyp x exp
N. 117: Y = A*EXP(B/X)/X <-- Hyp. x mod exp
N. 118: Y = A*X**(B*X**C)+D <-- Freundich + c
N. 119: Y = X/(A+B*X+C*SQRT(X))+D <-- Gunary + c
N. 120: Y = 1/(A+B*X**2)**C+D <-- Inv pow + c
N. 121: Y = A*(X**B)*Ln(X) <-- Pow x log
N. 122: Y = A*(X**B)/Ln(X) <-- Pow / log
N. 123: Y = A*(X**B)*Ln(X+C) <-- Pow x log (2)
N. 124: Y = A*X**B+C*EXP(D/X)+E <-- Pow + mod exp +c
N. 125: Y = A/X+B*X**C+D <-- Hyp + pow + c
N. 126: Y = A/X+B*EXP(C/X)+D <-- Hyp + mod exp + c
N. 127: Y = A/X+B*EXP(C*X)+D <-- Hyp + exp + c
N. 128: Y = A*B**X+C <--- Mod pow+ c
N. 129: Y = A*B**(1/X)+C <--- Root + c
N. 130: Y = A*X**(B*X)+C <--- Super geometric + c
N. 131: Y = A*X**(B/X)+C <--- Mod geometric + c
N. 132: Y = 1/(A+B*LnX)+C <--- Inv log + c
N. 133: Y = A*(Ln(X+B))**C <--- Pow of mod log
N. 134: Y = A*(Ln(X+B))**C+D <--- Pow of mod log + c
N. 135: Y = A*(B**X)*X**C+D <--- Hoerl + c
N. 136: Y = A*(B**(1/X))*X**C+D <--- Mod Hoerl + c
N. 137: Y = A*EXP(((LnX-B)**2)/C)+D <--- Log Gaussian + c
N. 138: Y = 1/(A*(X+B)**2+C)+D <--- Cauchy + c
N. 139: Y = A*((X/B)**C)*EXP(X/B)+D <--- Gamma + c
N. 140: Y = A*EXP(((X-B)**2)/C)+D <--- Gaussian + c
N. 141: Y = 1/(A+B*X+C/X)+D
N. 142: Y = A*X**(B+C*X)+D <--- Pow (2) + c
N. 143: Y = A*X**(B+C/X)+D <--- Pow (3) + c
N. 144: Y = A*X**(B+C*LnX)+D <--- Pow (4) + c
N. 145: Y = A*X**(B*X+C*X**2)+D <--- Pow (5) + c
N. 146: Y = A*EXP(B*X+C*X**0.5)+D <--- Exp (2) + c
N. 147: Y = A*EXP(B/X+C*X)+D <--- Exp (3) + c
N. 148: Y = (A+X)/(B+C*X)+D*X <--- Straight / straight + straight
N. 149: Y = (A+X)/(B+C*X**2)+D <--- Straight line / parabola + c
N. 150: Y = A*SENH(B*X) <--- Hyperbolical sine
N. 151: Y = A*COSH(B*X) <--- Hiperbolical cosine
N. 152: Y = SQRT(A-(X-B)**2)+C <--- circle + c
N. 153: Y = SQRT(A-B*(X-C)**2)+D <--- Ellipse + c
N. 154: Y = A*COSH(B*X) + C <--- Hiperbolical cossine + c
N. 155: Y = A*SENH(B*X) + C <--- Hyperbolical sine + c
N. 156: Y = SQRT(A-B*(X-C)**2)+D*X <--- Ellipse + Str
N. 157: Y = A*X**(B/X)+C*X <--- Mod geometric + Str
N. 158: Y = 1/(A+B*LnX)+C*X <--- Inv log + Str
N. 159: Y = A*(Ln(X+B))**C+D*X
N. 160: Y = A*(Ln(X+B))**C+D*X**2
N. 161: Y = A*(B**X)*X**C+D*X <--- Hoerl + Str
N. 162: Y = A*(B**(1/X))*X**C+D*X
N. 163: Y = A*EXP(((LnX-B)**2)/C)+D*X

N. 164: Y = 1/(A*(X+B)**2+C)+D*X <--- Cauchy + Str
N. 165: Y = A*((X/B)**C)*EXP(X/B)+D*X
N. 166: Y = A*EXP(((X-B)**2)/C)+D*X
N. 167: Y = 1/(A+B*X+C/X)+D*X
N. 168: Y = A*X**(B+C*X)+D*X <--- Pow (2) + Str
N. 169: Y = A*X**(B+C/X)+D*X <--- Pow (3) + Str
N. 170: Y = A*X**(B+C*LnX)+D*X <--- Pow (4) + Str
N. 171: Y = A*X**(B*X+C*X**2)+D*X <--- Pow (5) + Str
N. 172: Y = A*EXP(B*X+C*X**0.5)+D*X <--- Exp (2) + Str
N. 173: Y = A*EXP(B/X+C*X)+D*X <--- Exp (3) + Str
N. 174: Y = (A+X)/(B+C*X)+D*X**2 <--- Straight / straight + Parab
N. 175: Y = (A+X)/(B+C*X**2)+D*X
N. 176: Y = A*SENH(B*X)+C*X <--- Hyperbolical sine + Str
N. 177: Y = A*COSH(B*X)+C*X <--- Hiperbolical cossine + Str
N. 178: Y = SQRT(A-(X-B)**2)+C*X <--- circle + Str
N. 179: Y = SQRT(A-B*(X-C)**2)+D*X <--- Ellipse + Str
N. 180: Y = A*COSH(B*X)+C*X**2
N. 181: Y = A*SENH(B*X)+C*X**2
N. 182: Y = SQRT(A-B*(X-C)**2)+D*X**2
N. 183: Y = A*X**(B/X)+C*Ln(X)
N. 184: Y = 1/(A+B*LnX)+C*Ln(X)
N. 185: Y = A*(Ln(X+B))**C+D*Ln(X)
N. 186: Y = A*(Ln(X+B))**C+D*Ln(X)**2
N. 187: Y = A*(B**X)*X**C+D*Ln(X)
N. 188: Y = A*(B**(1/X))*X**C+D*Ln(X)
N. 189: Y = A*EXP(((LnX-B)**2)/C)+D*Ln(X)
N. 190: Y = 1/(A*(X+B)**2+C)+D*Ln(X)
N. 191: Y = A*((X/B)**C)*EXP(X/B)+D*Ln(X)
N. 192: Y = A*EXP(((X-B)**2)/C)+D*Ln(X)
N. 193: Y = 1/(A+B*X+C/X)+D*Ln(X)
N. 194: Y = A*X**(B+C*X)+D*Ln(X)
N. 195: Y = A*X**(B+C/X)+D*Ln(X)
N. 196: Y = A*X**(B+C*LnX)+D*Ln(X)
N. 197: Y = A*X**(B*X+C*X**2)+D*Ln(X)
N. 198: Y = A*EXP(B*X+C*X**0.5)+D*Ln(X)
N. 199: Y = A*EXP(B/X+C*X)+D*Ln(X)
N. 200: Y = (A+X)/(B+C*X)+D*Ln(X)**2
N. 201: Y = (A+X)/(B+C*X**2)+D*Ln(X)
N. 202: Y = A*SENH(B*X)+C*Ln(X)
N. 203: Y = A*COSH(B*X)+C*Ln(X)
N. 204: Y = SQRT(A-(X-B)**2)+C*Ln(X)
N. 205: Y = SQRT(A-B*(X-C)**2)+D*Ln(X)
N. 206: Y = A*COSH(B*X)+C*Ln(X)**2
N. 207: Y = A*SENH(B*X)+C*Ln(X)**2
N. 208: Y = SQRT(A-B*(X-C)**2)+D*Ln(X)**2

. . .

. . .

TECNICAL RESERVATION FOR NEW FUNCTIONS OF 1 INDEPENDENT VARIABLE

. . .

. . .

. . .

WITH 2 INDEPENDET VARIABLES

N. 300: Y=A*(X1**B)*X2**C <--- Pow x pow
N. 301: Y=X1/(A+B*X2) <--- Inv hyp (1)
N. 302: Y=X2/(A+B*X1) <--- Inv hyp (2)
N. 303: Y=A*X1**(B*X2) <--- Super geometric (1)
N. 304: Y=A*X2**(B*X1) <--- Super geometric (2)
N. 305: Y=A*X1**(B/X2) <--- Mod geometric (1)
N. 306: Y=A*X2**(B/X1) <--- Mod geometric (2)
N. 307: Y=A*X1+B*X2**2 <--- 2nd order (1)
N. 308: Y=A*X2+B*X1**2 <--- 2nd order (2)
N. 309: Y=X1/(A+B*X2**2) <--- Straight line / parabola (1)
N. 310: Y=X2/(A+B*X1**2) <--- Straight line / parabola (2)
N. 311: Y=A*(B**X1)*X2**C <--- Hoerl (1)
N. 312: Y=A*(B**X2)*X1**C <--- Hoerl (2)
N. 313: Y=A*(X1*X2)**B <--- Pow (1)
N. 314: Y=A*(X1/X2)**B <--- Pow (2)
N. 315: Y=A*(X1/X2)**B+C <--- Pow + c (3)
N. 316: Y=A*(B**(1/X1))*X2**C <--- Mod Hoerl (1)
N. 317: Y=A*(B**(1/X2))*X1**C <--- Mod Hoerl (2)
N. 318: Y=A+B/X1+C/X2**2 <--- 2nd order Hyperbola (1)
N. 319: Y=A+B/X2+C/X1**2 <--- 2nd order Hyperbola (2)
N. 320: Y=A+B*X1+C/X2 <--- Straight line + hyp (1)
N. 321: Y=A+B*X2+C/X1 <--- Straight line + hyp (2)
N. 322: Y=A*((X1/B)**C)*EXP(X2/B) <--- Gamma (1)
N. 323: Y=A*((X2/B)**C)*EXP(X1/B) <--- Gamma (2)
N. 324: Y=A+B*X1+C*X2**2 <--- Parabola (1)
N. 325: Y=A+B*X2+C*X1**2 <--- Parabola (2)
N. 326: Y=A*(X1**B)*X2**C+D <--- Pow x pow + c
N. 327: Y=X1/(A+B*X2)+C <--- Inv hyp + c (1)
N. 328: Y=X2/(A+B*X1)+C <--- Inv Hyp + c (2)
N. 329: Y=A*X1**(B*X2)+C <--- Super geometric + c (1)
N. 330: Y=A*X2**(B*X1)+C <--- Super geometric + c (2)
N. 331: Y=A*X1**(B/X2)+C <--- Mod geometric + c (1)
N. 332: Y=A*X2**(B/X1)+C <--- Mod geometric + c (2)
N. 333: Y=A*(B**(1/X1))*X2**C+D <--- Mod Hoerl + c (1)
N. 334: Y=A*(B**(1/X2))*X1**C+D <--- Mod Hoerl + c (2)
N. 335: Y=X1/(A+B*X2**2)+C <--- Straight line / parabola + c (1)
N. 336: Y=X2/(A+B*X1**2)+C <--- Straight line / parabola + c (2)
N. 337: Y=A*(B**X1)*X2**C+D <--- Hoerl + c (1)
N. 338: Y=A*(B**X2)*X1**C+D <--- Hoerl + c (2)
N. 339: Y=A*((X1/B)**C)*EXP(X2/B)+D <--- Gamma + c (1)
N. 340: Y=A*((X2/B)**C)*EXP(X1/B)+D <--- Gamma + c (2)
N. 341: Y=1/(A+B*X1+C/X2) <--- Inv: Straight line + hyp (1)
N. 342: Y=1/(A+B*X2+C/X1) <--- Inv: Straight line + hyp (1)
N. 343: Y=A+B*X1+C/X2**2 <--- Straight line + hyp (1)
N. 344: Y=A+B*X2+C/X1**2 <--- Straight line + hyp (1)
N. 345: Y=A*X1**(B+C*X2) <--- Pow (3)
N. 346: Y=A*X2**(B+C*X1) <--- Pow (4)
N. 347: Y=A*X1**(B+C/X2) <--- Pow (5)
N. 348: Y=A*X2**(B+C/X1) <--- Pow (6)
N. 349: Y=A*X1**(B+C*LnX2) <--- Pow (7)
N. 350: Y=A*X2**(B+C*LnX1) <--- Pow (8)
N. 351: Y=A*X2**(B+C/LnX1) <--- Pow (9)
N. 352: Y=A*X1**(B+C/LnX2) <-- Pow (10)
N. 353: Y=A*EXP(B*X1+C*X2**2) <-- Exp (1)

N. 354: Y=A*EXP(B*X2+C*X1**2) <-- Exp (2)
N. 355: Y=A*EXP(B/X1+C*X2) <-- Exp (3)
N. 356: Y=A*EXP(B/X2+C*X1) <-- Exp (4)
N. 357: Y=(A+X1)/(B+C*X2) <-- Straight line / Straight line (1)
N. 358: Y=(A+X2)/(B+C*X1) <-- Straight line / straight line (2)
N. 359: Y=(A+X1)/(B+C*X2**2) <-- Straight line / parabola (1)
N. 360: Y=(A+X2)/(B+C*X1**2) <-- Straight line / parabola (2)
N. 361: Y=A*EXP(B*X1)+C*EXP(D*X2) <-- 2 exp
N. 362: Y=A*(EXP(B*X1)-EXP(C*X2)) <-- Assoc exp
N. 363: Y=A*X1**B+C*X2**D <-- 2 pow
N. 364: Y=A*X1**B+C*EXP(D*X2) <-- Pow + exp (1)
N. 365: Y=A*X2**B+C*EXP(D*X1) <-- Pow + exp (2)
N. 366: Y=A*(X1**B)*(C-X2)**D <-- Mod Beta (1)
N. 367: Y=A*(X2**B)*(C-X1)**D <-- Mod Beta (2)
N. 368: Y=(A+B*X1**D)/(C+X2**D) <-- MMF (1)
N. 369: Y=(A+B*X2**D)/(C+X1**D) <-- MMF (2)
N. 370: Y=(A+B*X1)/(1+C*X2+D*X2**2) <-- Rational (1)
N. 371: Y=(A+B*X2)/(1+C*X1+D*X1**2) <-- Rational (2)
N. 372: Y=A*X1**(B*X2**C) <-- Freundich (1)
N. 373: Y=A*X2**(B*X1**C) <-- Freundich (2)
N. 374: Y=X1/(A+B*X2+C*SQRT(X2)) <-- Gunary (1)
N. 375: Y=X2/(A+B*X1+C*SQRT(X1)) <-- Gunary (2)
N. 376: Y=A*X1**B+C*EXP(D/X2) <-- Pow + mod exp (1)
N. 377: Y=A*X2**B+C*EXP(D/X1) <-- Pow + mod exp (2)
N. 378: Y=A*EXP(((X1-B)**2)/C+((X2-D)**2)/E) <-- Gaussian
N. 379: Y=A*EXP(((LnX1-B)**2)/C+((LnX2-D)**2)/E)
N. 380: Y=A*X2**2+B*X2+C*X1+D
N. 381: Y=A*X1**2+B*X1+C*X2+D
N. 382: Y=A*X1**3+B*X1**2+C*X1+D*X2
N. 383: Y=A*X2**3+B*X2**2+C*X2+D*X1
N. 384: Y=A*X2**3+B*X2**2+C*X2+D*X1+E
N. 385: Y=A*X1**3+B*X1**2+C*X1+D*X2+E
N. 386: Y=EXP(A+B/X1+C*Ln(X2)) <-- Vapor Pressure (1)
N. 387: Y=EXP(A+B/X2+C*Ln(X1)) <-- Vapor Pressure (2)
N. 388: Y=EXP(A+B/X1+C*Ln(X2))+D <-- Vapor Pressure (3)
N. 389: Y=EXP(A+B/X2+C*Ln(X1))+D <-- Vapor Pressure (4)
N. 390: Y=A*(X1**B)*Ln(X2+C) <-- Pow x log (1)
N. 391: Y=A*(X2**B)*Ln(X1+C) <-- Pow x log (2)
N. 392: Y=A*(X1**B)*Ln(X2+C)+D <-- Pow x log (3)
N. 393: Y=A*(X2**B)*Ln(X1+C)+D <-- Pow x log (4)
N. 394: Y=A/X1+B*EXP(C/X2)+D <-- Hyp + mod exp (1)
N. 395: Y=A/X2+B*EXP(C/X1)+D <-- Hyp + mod exp (2)
N. 396: Y=A/X1+B*EXP(C*X2)+D <-- Hyp + exp (1)
N. 397: Y=A/X2+B*EXP(C*X1)+D <-- Hyp + exp (2)
N. 398: Y=A*X1**(B+C*X2)+D <-- Pow (3) + c
N. 399: Y=A*X2**(B+C*X1)+D <-- Pow (4) + c
N. 400: Y=A*X1**(B+C/X2)+D <-- Pow (5) + c
N. 401: Y=A*X2**(B+C/X1)+D <-- Pow (6) + c
N. 402: Y=A*X1**(B+C*LnX2)+D <-- Pow (7) + c
N. 403: Y=A*X2**(B+C*LnX1)+D <-- Pow (8) + c
N. 404: Y=A*X2**(B+C/LnX1)+D <-- Pow (9) + c
N. 405: Y=A*X1**(B+C/LnX2)+D <-- Pow (10) + c
N. 406: Y=A*EXP(B*X1+C*X2**2)+D <-- Exp (1) + c
N. 407: Y=A*EXP(B*X2+C*X1**2)+D <-- Exp (2) + c
N. 408: Y=A*EXP(B/X1+C*X2)+D <-- Exp (3) + c
N. 409: Y=A*EXP(B/X2+C*X1)+D <-- Exp (4) + c

N. 410: Y=(A+X1)/(B+C*X2)+D <-- Str / Str (1) + c
N. 411: Y=(A+X2)/(B+C*X1)+D <-- Str / Str (2) + c
N. 412: Y=(A+X1)/(B+C*X2**2)+D <-- Str / Parab (1) + c
N. 413: Y=(A+X2)/(B+C*X1**2)+D <-- Str / Parab (2) + c
N. 414: Y=A*X1**(B*X2**C)+D <-- Freundich (1) + c
N. 415: Y=A*X2**(B*X1**C)+D <-- Freundich (2) + c
N. 416: Y=A*X1+B*X2+C <-- Str + Str
N. 417: Y=A*X1+B*X1**2+C*X2+D*X2**2 <-- Parab + Parab
N. 418: Y=A*(X1**B)*Ln(X2+C)+D*X1
N. 419: Y=A*(X2**B)*Ln(X1+C)+D*X1
N. 420: Y=A/X1+B*EXP(C/X2)+D*X1
N. 421: Y=A/X2+B*EXP(C/X1)+D*X1
N. 422: Y=A/X1+B*EXP(C*X2)+D*X1
N. 423: Y=A/X2+B*EXP(C*X1)+D*X1
N. 424: Y=A*X1**(B+C*X2)+D*X1
N. 425: Y=A*X2**(B+C*X1)+D*X1
N. 426: Y=A*X1**(B+C/X2)+D*X1
N. 427: Y=A*X2**(B+C/X1)+D*X1
N. 428: Y=A*X1**(B+C*LnX2)+D*X1
N. 429: Y=A*X2**(B+C*LnX1)+D*X1
N. 430: Y=A*X2**(B+C/LnX1)+D*X1
N. 431: Y=A*X1**(B+C/LnX2)+D*X1
N. 432: Y=A*EXP(B*X1+C*X2**2)+D*X1
N. 433: Y=A*EXP(B*X2+C*X1**2)+D*X1
N. 434: Y=A*EXP(B/X1+C*X2)+D*X1
N. 435: Y=A*EXP(B/X2+C*X1)+D*X1
N. 436: Y=(A+X1)/(B+C*X2)+D*X1
N. 437: Y=(A+X2)/(B+C*X1)+D*X1
N. 438: Y=(A+X1)/(B+C*X2**2)+D*X1
N. 439: Y=(A+X2)/(B+C*X1**2)+D*X1
N. 440: Y=A*X1**(B*X2**C)+D*X1
N. 441: Y=A*X2**(B*X1**C)+D*X1
N. 442: Y=A*X1+B*X2+C*X1**4
N. 443: Y=A*X1+B*X1**3+C*X2+D*X2**3 <-- Parab + Parab
N. 444: Y=A*(X1**B)*Ln(X2+C)+D*X2
N. 445: Y=A*(X2**B)*Ln(X1+C)+D*X2
N. 446: Y=A/X1+B*EXP(C/X2)+D*X2
N. 447: Y=A/X2+B*EXP(C/X1)+D*X2
N. 448: Y=A/X1+B*EXP(C*X2)+D*X2
N. 449: Y=A/X2+B*EXP(C*X1)+D*X2
N. 450: Y=A*X1**(B+C*X2)+D*X2
N. 451: Y=A*X2**(B+C*X1)+D*X2
N. 452: Y=A*X1**(B+C/X2)+D*X2
N. 453: Y=A*X2**(B+C/X1)+D*X2
N. 454: Y=A*X1**(B+C*LnX2)+D*X2
N. 455: Y=A*X2**(B+C*LnX1)+D*X2
N. 456: Y=A*X2**(B+C/LnX1)+D*X2
N. 457: Y=A*X1**(B+C/LnX2)+D*X2
N. 458: Y=A*EXP(B*X1+C*X2**2)+D*X2
N. 459: Y=A*EXP(B*X2+C*X1**2)+D*X2
N. 460: Y=A*EXP(B/X1+C*X2)+D*X2
N. 461: Y=A*EXP(B/X2+C*X1)+D*X2
N. 462: Y=(A+X1)/(B+C*X2)+D*X2
N. 463: Y=(A+X2)/(B+C*X1)+D*X2
N. 464: Y=(A+X1)/(B+C*X2**2)+D*X2
N. 465: Y=(A+X2)/(B+C*X1**2)+D*X2

N. 466: Y=A*X1**(B*X2**C)+D*X2
N. 467: Y=A*X2**(B*X1**C)+D*X2
N. 468: Y=A*X1+B*X2+C*X2**4
N. 469: Y=A*X1+B/X1**3+C*X2+D/X2**3
N. 470: Y=A*(X1**B)*Ln(X2+C)+D/X2
N. 471: Y=A*(X2**B)*Ln(X1+C)+D/X2
N. 472: Y=A/X1+B*EXP(C/X2)+D/X2
N. 473: Y=A/X2+B*EXP(C/X1)+D/X2**2
N. 474: Y=A/X1+B*EXP(C*X2)+D/X2
N. 475: Y=A/X2+B*EXP(C*X1)+D/X2**2
N. 476: Y=A*X1**(B+C*X2)+D/X2
N. 477: Y=A*X2**(B+C*X1)+D/X2
N. 478: Y=A*X1**(B+C/X2)+D/X2
N. 479: Y=A*X2**(B+C/X1)+D/X2
N. 480: Y=A*X1**(B+C*LnX2)+D/X2
N. 481: Y=A*X2**(B+C*LnX1)+D/X2
N. 482: Y=A*X2**(B+C/LnX1)+D/X2
N. 483: Y=A*X1**(B+C/LnX2)+D/X2
N. 484: Y=A*EXP(B*X1+C*X2**2)+D/X2
N. 485: Y=A*EXP(B*X2+C*X1**2)+D/X2
N. 486: Y=A*EXP(B/X1+C*X2)+D/X2
N. 487: Y=A*EXP(B/X2+C*X1)+D/X2
N. 488: Y=(A+X1)/(B+C*X2)+D/X2
N. 489: Y=(A+X2)/(B+C*X1)+D/X2
N. 490: Y=(A+X1)/(B+C*X2**2)+D/X2
N. 491: Y=(A+X2)/(B+C*X1**2)+D/X2
N. 492: Y=A*X1**(B*X2**C)+D/X2
N. 493: Y=A*X2**(B*X1**C)+D/X2
N. 494: Y=A*X1+B*X2+C/X2**4
N. 495: Y=A*X1+B/X1**2+C*X2+D/X2**2
N. 496: Y=A*(X1**B)*Ln(X2+C)+D/X1
N. 497: Y=A*(X2**B)*Ln(X1+C)+D/X1
N. 498: Y=A/X1+B*EXP(C/X2)+D/X1**2
N. 499: Y=A/X2+B*EXP(C/X1)+D/X1
N. 500: Y=A/X1+B*EXP(C*X2)+D/X1**2
N. 501: Y=A/X2+B*EXP(C*X1)+D/X1
N. 502: Y=A*X1**(B+C*X2)+D/X1
N. 503: Y=A*X2**(B+C*X1)+D/X1
N. 504: Y=A*X1**(B+C/X2)+D/X1
N. 505: Y=A*X2**(B+C/X1)+D/X1
N. 506: Y=A*X1**(B+C*LnX2)+D/X1
N. 507: Y=A*X2**(B+C*LnX1)+D/X1
N. 508: Y=A*X2**(B+C/LnX1)+D/X1
N. 509: Y=A*X1**(B+C/LnX2)+D/X1
N. 510: Y=A*EXP(B*X1+C*X2**2)+D/X1
N. 511: Y=A*EXP(B*X2+C*X1**2)+D/X1
N. 512: Y=A*EXP(B/X1+C*X2)+D/X1
N. 513: Y=A*EXP(B/X2+C*X1)+D/X1
N. 514: Y=(A+X1)/(B+C*X2)+D/X1
N. 515: Y=(A+X2)/(B+C*X1)+D/X1
N. 516: Y=(A+X1)/(B+C*X2**2)+D/X1
N. 517: Y=(A+X2)/(B+C*X1**2)+D/X1
N. 518: Y=A*X1**(B*X2**C)+D/X1
N. 519: Y=A*X2**(B*X1**C)+D/X1
N. 520: Y=A*X1+B*X2+C/X1**4
N. 521: Y=A*X1+B/X1**3+C*X2+D*Ln(X2)**3

N. 522: Y=A*(X1**B)*Ln(X2+C)+D*Ln(X2)
N. 523: Y=A*(X2**B)*Ln(X1+C)+D*Ln(X2)
N. 524: Y=A/X1+B*EXP(C/X2)+D*Ln(X2)
N. 525: Y=A/X2+B*EXP(C/X1)+D*Ln(X2)**2
N. 526: Y=A/X1+B*EXP(C*X2)+D*Ln(X2)
N. 527: Y=A/X2+B*EXP(C*X1)+D*Ln(X2)**2
N. 528: Y=A*X1**(B+C*X2)+D*Ln(X2)
N. 529: Y=A*X2**(B+C*X1)+D*Ln(X2)
N. 530: Y=A*X1**(B+C/X2)+D*Ln(X2)
N. 531: Y=A*X2**(B+C/X1)+D*Ln(X2)
N. 532: Y=A*X1**(B+C*LnX2)+D*Ln(X2)
N. 533: Y=A*X2**(B+C*LnX1)+D*Ln(X2)
N. 534: Y=A*X2**(B+C/LnX1)+D*Ln(X2)
N. 535: Y=A*X1**(B+C/LnX2)+D*Ln(X2)
N. 536: Y=A*EXP(B*X1+C*X2**2)+D*Ln(X2)
N. 537: Y=A*EXP(B*X2+C*X1**2)+D*Ln(X2)
N. 538: Y=A*EXP(B/X1+C*X2)+D*Ln(X2)
N. 539: Y=A*EXP(B/X2+C*X1)+D*Ln(X2)
N. 540: Y=(A+X1)/(B+C*X2)+D*Ln(X2)
N. 541: Y=(A+X2)/(B+C*X1)+D*Ln(X2)
N. 542: Y=(A+X1)/(B+C*X2**2)+D*Ln(X2)
N. 543: Y=(A+X2)/(B+C*X1**2)+D*Ln(X2)
N. 544: Y=A*X1**(B*X2**C)+D*Ln(X2)
N. 545: Y=A*X2**(B*X1**C)+D*Ln(X2)
N. 546: Y=A*X1+B*X2+C*Ln(X2)**4
N. 547: Y=A*X1+B/X1**2+C*X2+D*Ln(X2)**2
N. 548: Y=A*(X1**B)*Ln(X2+C)+D*Ln(X1)
N. 549: Y=A*(X2**B)*Ln(X1+C)+D*Ln(X1)
N. 550: Y=A/X1+B*EXP(C/X2)+D*Ln(X1)**2
N. 551: Y=A/X2+B*EXP(C/X1)+D*Ln(X1)
N. 552: Y=A/X1+B*EXP(C*X2)+D*Ln(X1)**2
N. 553: Y=A/X2+B*EXP(C*X1)+D*Ln(X1)
N. 554: Y=A*X1**(B+C*X2)+D*Ln(X1)
N. 555: Y=A*X2**(B+C*X1)+D*Ln(X1)
N. 556: Y=A*X1**(B+C/X2)+D*Ln(X1)
N. 557: Y=A*X2**(B+C/X1)+D*Ln(X1)
N. 558: Y=A*X1**(B+C*LnX2)+D*Ln(X1)
N. 559: Y=A*X2**(B+C*LnX1)+D*Ln(X1)
N. 560: Y=A*X2**(B+C/LnX1)+D*Ln(X1)
N. 561: Y=A*X1**(B+C/LnX2)+D*Ln(X1)
N. 562: Y=A*EXP(B*X1+C*X2**2)+D*X1
N. 563: Y=A*EXP(B*X2+C*X1**2)+D*Ln(X1)
N. 564: Y=A*EXP(B/X1+C*X2)+D*Ln(X1)
N. 565: Y=A*EXP(B/X2+C*X1)+D*Ln(X1)
N. 566: Y=(A+X1)/(B+C*X2)+D*Ln(X1)
N. 567: Y=(A+X2)/(B+C*X1)+D*Ln(X1)
N. 568: Y=(A+X1)/(B+C*X2**2)+D*Ln(X1)
N. 569: Y=(A+X2)/(B+C*X1**2)+D*Ln(X1)
N. 570: Y=A*X1**(B*X2**C)+D*Ln(X1)
N. 571: Y=A*X2**(B*X1**C)+D*Ln(X1)
N. 572: Y=A*X1+B*X2+C*Ln(X1)**4