Fall 2006

Quick Tour of Mathcad
and Examples

Mathcad provides a unique and powerful way to work with equations, numbers, tests and graphs.

Features

- Arithmetic
- Functions
- Plot functions
- Define you own variables and functions
- Visualize data in 2-D and 3-D
- Compute sums and integrals
- Do matrix computations
- Solve equations
  - numerically
  - symbolically
- Programming
- Curve fitting and data analysis functions

Arithmetic, variables, constants, units, equal signs.

The **=** sign means to print the value

The **:** sign means to define a variable [You type : but mathcad prints :=]

The **~** sign means to define a variable
before the rest of the work sheet is evaluated

The <cntrl>=, ■ = ■, means to set equal "logically", that is without numerical evaluation.

The **..** range operator defines the range in the toolbar or use a semicolon (**;**)

i= 0, 0.5 .. 10

Typing @ generates a blank graph **or** use the X-Y Plot button in **Graph (Insert or Toolbar)**.

To separate expressions use the comma operator (**,**).

Arithmetic, Functions, Calculus, Greek letters, Matrix, Symbolic, Programming etc ...
can be selected in the Toolbar. (**View, then Toolbar**)

(a). Arithmetic To the right, type the following (NOT IN A TEXT REGION) [For tips on editing and typing expressions, see **Editing** below.] To add text Go to top menu and **Insert** and select **Text Region**

2/3=

2*3=
2^3=
compare
2/3:

(b). Now type the following:
x:2
y:3
x/y=
x*y=
x^y=
(c). Mathcad has several built-in constants. That is, Mathcad "knows" these constants.

Find the values of the following constants by typing
e=
p<Ctrl-g>=
[This means type p and hold down the Ctrl key while you press g.]

R=

Note that R is NOT the gas constant.

Editing: Correcting a typing error: place the mouse just to the right of the wrong character or symbol.
Click and press the delete (back arrow) key. Try this: change 3 to 9 in the math expressions you created at left.

Editing:
Whole mathcad expressions or text regions can be removed. Just select the desired region and press the delete or backspace key. There are two ways to select a region. Either click and drag into the region, or click once in the region and press <up arrow> repeating until whole region is outlined.

Editing:
Expressions can be edited for variables and constants or for operators. To remove (and replace) and operator, click on the expression and use the up-arrow or down-arrow to select the part of the expression that contains the offending operator. (blue editing line)
Press the back-arrow to remove it and press the desired operator key to replace it.

Arithmetic:

$\frac{197}{13} = 15.154$ $\qquad 13^7 = 6.275 \times 10^7$ $\qquad \sqrt{\frac{1.837 \cdot 10^2}{50 + 6^5}} = 0.153$

$$\prod_{j=1}^{5}\sum_{i=1}^{j}\frac{1}{i} = 13.082$$

Functions:

$$\log(145)\cdot\cos\left(\frac{3}{5}\cdot\pi\right) = -0.668$$

Complex numbers

$$(5.1+4i)^2 + e^{3-2i} = 1.651 + 22.536i$$

Units: (to check or add new units use top menu Insert and Units)

$$\frac{1250\cdot km}{1.5\cdot hr} = 231.481\,\frac{m}{s}$$

change units directly to mph

Function and Plot:

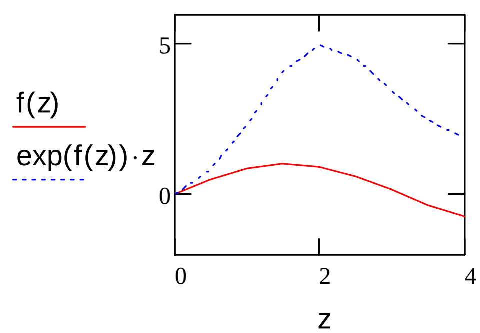
$z := 0, 0.5 .. 4$ $\qquad f(z) := \sin(z)$

$z =$

0.000
0.500
1.000
1.500
2.000
2.500
3.000
3.500
4.000

$f(z) =$

0.000
0.479
0.841
0.997
0.909
0.598
0.141
-0.351
-0.757



Variables and your own functions:

$a := 4$ $\qquad a\cdot 5 + \sqrt[3]{a} = 21.587$

Or $\qquad f(x) := \dfrac{\sin(x)}{\dfrac{x}{a^3}}$

$f(1) = 53.854$ $\qquad f(10) = -3.482$

Calculus:

$$\sum_{n=0}^{12} \frac{1}{n!} = 2.718 \qquad \int_0^1 \frac{1}{1+\sqrt[3]{x}}\,dx = 0.579$$

Matrix

$$A := \begin{pmatrix} 4 & 5 & 1 \\ 5 & 0 & -2 \\ -1 & 2 & 8 \end{pmatrix}$$

Inverse matrix $A^{-1} = \begin{pmatrix} -0.024 & 0.232 & 0.061 \\ 0.232 & -0.201 & -0.079 \\ -0.061 & 0.079 & 0.152 \end{pmatrix}$

$$A \cdot A^{-1} = \begin{pmatrix} 1.000 & -6.939\times 10^{-17} & -2.776\times 10^{-17} \\ 1.388\times 10^{-17} & 1.000 & 0.000 \\ 5.551\times 10^{-17} & 0.000 & 1.000 \end{pmatrix}$$

determinant

$|A| = -164.000$

$$B := \begin{pmatrix} 1 & 2 \\ 4 & 6 \end{pmatrix} \qquad C := \begin{pmatrix} 3 & 7 \\ -3 & -2 \end{pmatrix}$$

$$B^2 = \begin{pmatrix} 9.000 & 14.000 \\ 28.000 & 44.000 \end{pmatrix}$$

$$B \cdot C = \begin{pmatrix} -3.000 & 3.000 \\ -6.000 & 16.000 \end{pmatrix} \qquad B - C = \begin{pmatrix} -2.000 & -5.000 \\ 7.000 & 8.000 \end{pmatrix} \qquad \frac{B}{C} = \begin{pmatrix} 0.267 & -0.067 \\ 0.667 & -0.667 \end{pmatrix}$$

Solving Equations:(numerically, symbolically, using Symbolics, Root, Polyrrot and Given...Find build in functions)

Numericall

$t := 3$

$$\mathrm{root}\left(t^2 - \cosh(t), t\right) = 2.594$$

Giver

$v := 1$

$$v^2 + 3v - 5 = 0$$

$$\mathrm{Find}(v) = 1.193$$

Symbolically using "solve"

$$x1 + 1 = \frac{1}{x1} \text{ solve}, x1 \rightarrow \begin{pmatrix} \frac{1}{2}\cdot 5^{\frac{1}{2}} - \frac{1}{2} \\ \frac{-1}{2} - \frac{1}{2}\cdot 5^{\frac{1}{2}} \end{pmatrix}$$

$$\sum_{i=1}^{n} i^2 \rightarrow \frac{1}{3}\cdot(n+1)^3 - \frac{1}{2}\cdot(n+1)^2 + \frac{1}{6}\cdot n + \frac{1}{6}$$

change the value of **n** in the Sum symbol

Precautions

1. Numerical formatting: Under Format and Result there are three parameters that you have to be careful about.

(a) Under **Number Format--General** (Number of Decimal Places and Exponential Threshold)

Number of decimal places determines the number of figures displayed after the decimal (3 default).
Exponential Threshold determines how large or how small a number can become before the program switches to exponential notation (3 default)
(b) under **Number Format ,** Tolerence determines the size of a number that will be displayed as 0

decimal = 9 and
zero threshold = 9

$h := 6.626075\cdot 10^{-34}$

$h3 := h^3$ $\qquad h3 = 2.909 \times 10^{-100}$

then change
zero threshold to 150

$10^{10} h3 = 2.909 \times 10^{-90}$

Largest and smallest numbers are 10^{307} and 10^{-307}

2. Errors:

R is Rankine-scale degree temperature NOT the gas constant
g: acceleration of gravity NOT gram
Second is sec NOT s
Meter (unit of lenght SI system of unit) is m. In general you should not use (m, sec, gm and K) as variable names.

Kelvin temperature is denoted K therefore if you need to use it as equilibrium constant add a subscript K_{eq} for instance.

Mathcad evaluates expression from left to right and top to bottom.

Each sheet is indenpendent of the others.

Default numbering system for subscripts begins with 0 not 1. x_0, x_1, ...

Fitting Functions

I. Linear Regression:

Suppose the data consists of paired points $\{x_i\}$, $\{y_i\}$. The regression refers to finding the coefficient c_i of the following equation and to minimize the sum of squares of the deviations of the calculated and observed y.

$$y = \sum_i \left[c_i \cdot f_i(x) \right]$$

where c_i and $f_i(x)$ are the coefficient and arbitrary functions of x, respectively.

"Fitting data to a linear function" of the form y = ax + b where a is the slope and b the intercept.

Example 1:

You need first to define the number of points N and then the range variable i (index):

$N := 10 \qquad i := 0..N$

Data: $x_i :=$ $\quad y_i :=$

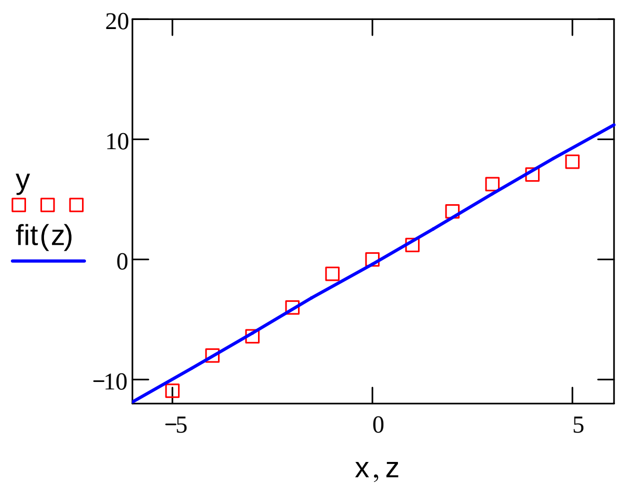
x_i	y_i
−5	−11
−4	−8
−3	−6.5
−2	−4
−1	−1.3
0	0
1	1.1
2	4
3	6.2
4	7
5	8.1

$a := \text{slope}(x, y) \qquad a = 1.927$

$b := \text{intercept}(x, y) \qquad b = -0.400$

$\text{fit}(z) := a \cdot z + b$

$z := -6, -4.5 .. 6$



II. Least Squares Fits:

Suppose the dataset is $\{(x_i,y_i)$ with $i = 1,2,3, ... n\}$. And you need to find the best polynomial fit to these data points in the form of f(z):

$$f(z) = \sum_{j=0}^{d} \left(c_j \cdot z^j\right) \qquad \text{equivalent to} \qquad f(z) = c_0 + c_1 \cdot z + c_2 \cdot z^2 + c_3 \cdot z^3$$

with d = 3

The best fit of y(x) to the data is found by minimizing the squared deviation function:

equivalent to

$$M = \sum_{i=1}^{n} \left[y_i - \sum_{j=0}^{d} \left[c_j \cdot (x_i)^j \right] \right]^2 \qquad M = \left(y_1 - c_0 + c_1 \cdot x_1\right)^2 + \left(y_2 - c_0 + c_1 \cdot x_2\right)^2$$

with n = 2 and d=1

Define a function X such as $X_{i,j} = (x_i)^j$

and a matrix M such as $M = X^T \cdot X$

Then the values of c_j that minimize M are given by

$$c = M^{-1} \cdot X \cdot y$$

Example 2: $x := (-2.5 \quad -1.0 \quad -0.2 \quad 0.5 \quad 2.2 \quad 3.9 \quad 5.6)^T$

$y := (5.2 \quad 2.9 \quad 1.1 \quad 1.6 \quad 5.75 \quad 18 \quad 29)^T$

$n := \text{length}(x)$ ***or*** $n := \text{rows}(x)$ $\quad i := 0 .. n-1 \quad d := 2 \quad j := 0 .. d$

Using the functions defined above $X_{i,j} := (x_i)^j$ and $M := X^T \cdot X$

$$M = \begin{pmatrix} 7.000 & 8.500 & 58.950 \\ 8.500 & 58.950 & 229.075 \\ 58.950 & 229.075 & 1.278 \times 10^{3} \end{pmatrix}$$

$$M^{-1} = \begin{pmatrix} 0.247 & 0.029 & -0.017 \\ 0.029 & 0.059 & -0.012 \\ -0.017 & -0.012 & 3.679 \times 10^{-3} \end{pmatrix}$$

$X_{i,j} =$

1.000
1.000
1.000
1.000
1.000
1.000
1.000
-2.500

$$X = \begin{pmatrix} 1.000 & -2.500 & 6.250 \\ 1.000 & -1.000 & 1.000 \\ 1.000 & -0.200 & 0.040 \\ 1.000 & 0.500 & 0.250 \\ 1.000 & 2.200 & 4.840 \\ 1.000 & 3.900 & 15.210 \\ 1.000 & 5.600 & 31.360 \end{pmatrix}$$

---

example: 3x3 matrix

$$\left(A^{T}\right)^{T} = \begin{pmatrix} 4.000 & 5.000 & 1.000 \\ 5.000 & 0.000 & -2.000 \\ -1.000 & 2.000 & 8.000 \end{pmatrix} \qquad A^{-1} = \begin{pmatrix} -0.024 & 0.232 & 0.061 \\ 0.232 & -0.201 & -0.079 \\ -0.061 & 0.079 & 0.152 \end{pmatrix}$$

$$A^{T} = \begin{pmatrix} 4.000 & 5.000 & -1.000 \\ 5.000 & 0.000 & 2.000 \\ 1.000 & -2.000 & 8.000 \end{pmatrix}$$

$|A| = -164.000$ nonsingular (invertible) determinant is not zero

---

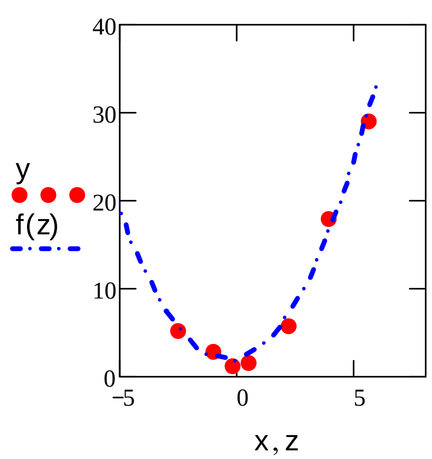
Back to our problem, the coefficients c are given by

$$c := M^{-1} \cdot X^{T} \cdot y \qquad c = \begin{pmatrix} 1.681 \\ 0.556 \\ 0.798 \end{pmatrix}$$

Polynomial fitting function:

$$f(z) := \sum_{j=0}^{d} \left(c_j \cdot z^j\right)$$

$z := -5, -4.9 .. 8$



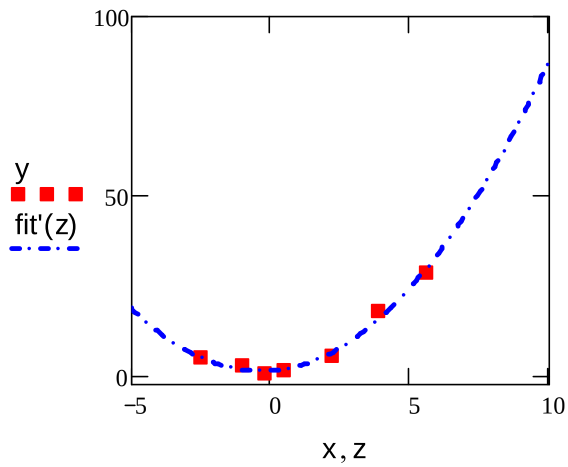
***Use of "linfit" function* (MathCAD built in function)**

Here $y(x) = c_o + c_1\, x + c_2\, x^2$. We need to define a function Y(x) which gives the power of the variable x.

The *linfit(x,y,Y)* function returns a vector containing the parameters used to create a linear combination of the functions in vector Y that best approximates the data in x and y in the least-squares sense.There must be at least as many data points as there are terms in Y.
Y(x) is a vector of functions; each element is one linear functional term in the fit function. In the case of a single linear function, Y is a scalar.

$$Y(x) := \begin{pmatrix} 1 \\ x \\ x^2 \end{pmatrix} \qquad c' := \text{linfit}(x, y, Y) \qquad c' = \begin{pmatrix} 1.681 \\ 0.556 \\ 0.798 \end{pmatrix} \qquad z := -5, -4.9 .. 10$$

$$\text{fit}'(z) := Y(z) \cdot \text{linfit}(x, y, Y)$$



More functions are defined in MathCAD

1. **General least square such as genfit, linfit , regress with interp.**

2. **Specialized least square functions such expfit, lnfit, logfit, pwrfit, sinfit etc ...**

Logarithmic Regression

The use of Mathcad’s **linfit** function to fit data to a logarithmic model is illustated bellow. Two specialized built-in fitting functions, **logfit** and **lnfit**, may also be used.
The following **Data Table** contains a set of data that may be best approximated by a logarithmic model function.

data :=

	0	1
0	1	4.18
1	2	4.67
2	3	5.3
3	4	5.37
4	5	5.45
5	6	5.74
6	7	5.65
7	8	5.84
8	9	6.36
9	10	6.38

$X := data^{\langle 0 \rangle}$ $\qquad Y := data^{\langle 1 \rangle}$

Enter the vector of functions to fit. In this example, we fit this data to the model function $y = a \cdot \ln(x) + b \cdot \sqrt{x} + c$, where a, b, and c are unknown.

$$F(x) := \begin{pmatrix} \ln(x) \\ \sqrt{x} \\ 1 \end{pmatrix}$$

1. Call the ***linfit*** function using the data and the vector of functions, *F*.

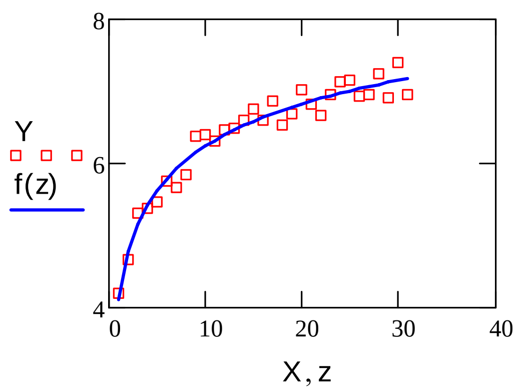
$$S1 := \text{linfit}(X, Y, F)$$

Define a function that uses these newly found parameter values in the logarithmic model. Also, define a range variable over which to graph the function.

$f(x) := F(x) \cdot S1$

$z := \min(X) .. \max(X)$ ***(range variable for plotting)***

A graph of the model function with the newly found parameter values and the original data points reveals a good fit.



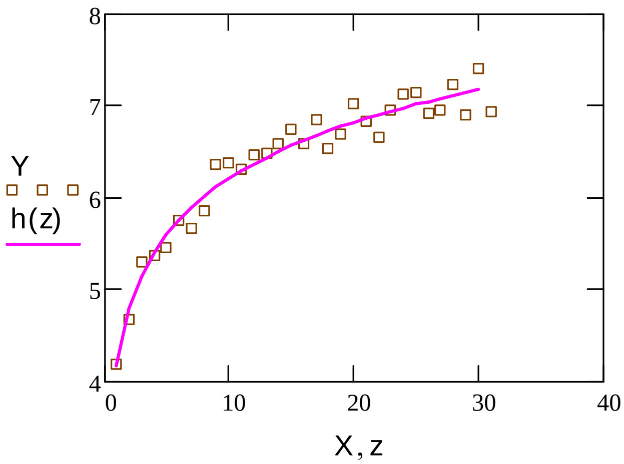
$\text{corr}\left(\overrightarrow{f(X)}, Y\right) = 0.979$ correlation coefficient

2. A simpler way to perform a customized *log fit* is by using the *lnfit* function.
The data is fitted with the model function $y = a \cdot \ln(x) + b$:

$T1 := \mathrm{lnfit}(X, Y)$ $\qquad T1 = \begin{pmatrix} 0.881 \\ 4.173 \end{pmatrix}$

$h(x) := T1_0 \cdot \ln(x) + T1_1$

$z := 0 .. \mathrm{last}(X)$ ***(range variable for plotting), a different way to set a range of values***



$\mathrm{corr}\left(\overrightarrow{h(X)}, Y\right) = 0.979$ correlation coefficient

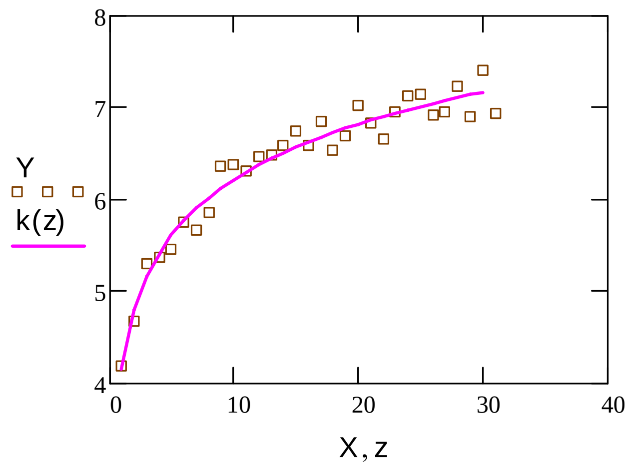
3. You can also use Mathcad's *logfit* function, which fits data to the model function.
The model function used here is $y = a \cdot \ln(x + b) + c$.

$Q := \begin{pmatrix} 1 \\ 0 \\ 4 \end{pmatrix}$ guess vector

$M := \mathrm{logfit}(X, Y, Q)$ $\qquad M = \begin{pmatrix} 0.864 \\ -0.097 \\ 4.227 \end{pmatrix}$

$k(x) := M_0 \cdot \ln(x + M_1) + M_2$

$z := 0 .. \text{last}(X)$ range variable for plotting



$\text{corr}\left(\overrightarrow{k(X)}, Y\right) = 0.979$ correlation coefficient

III. Polynomial Regression:

Polynomial function that best fit the set of data points

The function **regress(X,Y,*k*)** returns a vector ***s*** which **interp** uses to find the ***k*****th** order polynomial that best fits the ***x*** and ***y*** data values.

The function **interp(*s*,X,Y,*x*)** returns interpolated **y** value corresponding to ***x***.

X is a vector of real data values in ascending order (the ***x*** values).
Y is a vector of real data values (the ***y*** values).
s is a vector generated by the function regress.
k is a positive integer specifying the order of the polynomial you want to use to fit the data. Usually you'll want ***k* < 5**.
x is the value of the independent variable at which you want to evaluate the regression curve.

Notes:

•***Regress*** is useful when you have a set of measured ***y*** values corresponding to ***x*** values and you want to fit a single polynomial of any order to those ***y*** values.
• You should always use ***interp*** after using the ***regress*** function.
• Since ***regress*** tries to accommodate all your data points using a single polynomial, it will not work well when your data does not behave like a single polynomial.

Example *: Consider a matrix of X-Y data to be analyzed (X coordinate in first column, Y coordinate in second)*

$$\text{data} := \begin{pmatrix} 0 & 9.1 \\ 1 & 7.3 \\ 2 & 3.2 \\ 3 & 4.6 \\ 4 & 4.8 \\ 5 & 2.9 \\ 6 & 5.7 \\ 7 & 7.1 \\ 8 & 8.8 \\ 9 & 10.2 \end{pmatrix}$$

$$X := \text{data}^{\langle 0 \rangle}$$
$$Y := \text{data}^{\langle 1 \rangle}$$
$$n := \text{rows}(\text{data}) \qquad n = 10.000$$

Enter the degree of polynomial to fit, value of k: $k := 3$

*Define the polynomial function that fit best the set of data points using "**regress**" as follows:*

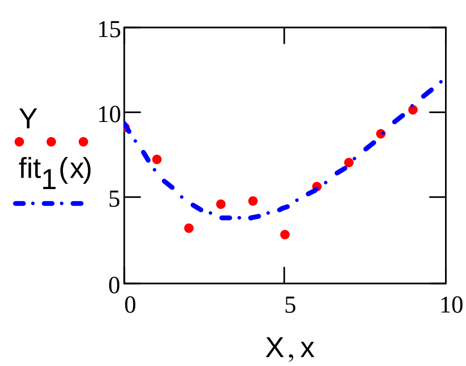
$$s_1 := \text{regress}(X, Y, k)$$

$$s_1 = \begin{pmatrix} 3.000 \\ 3.000 \\ 3.000 \\ 9.298 \\ -3.438 \\ 0.609 \\ -0.024 \end{pmatrix}$$

*Use the function "**interp**" as the polynomial fitting function:*

$$\text{fit}_1(x) := \text{interp}\left(s_1, X, Y, x\right)$$ *interpolation of **y** value corresponding to **x** to give a vector s*

$$x := 0, 1 .. 10$$



$$coeffs := submatrix(s_1, 3, length(s_1) - 1, 0, 0)$$

$$coeffs = \begin{pmatrix} 9.298 \\ -3.438 \\ 0.609 \\ -0.024 \end{pmatrix}$$

The coefficients are given by:

$$coeffs^T = \begin{pmatrix} 9.298 & -3.438 & 0.609 & -0.024 \end{pmatrix}$$ (matrix transpose)

$$Fit(x) := \sum_{i=0}^{k} \left(coeffs_i \cdot x^i\right)$$

cspline, pspline and lspline and **interp** functions for connecting X-Y data.

Enter a matrix of X-Y data to be interpolated:

$M_1 :=$

	0	1
0	0	0
1	1	3
2	4	27.57
3	5	25
4	7	18
5	8	30.41
6	11	48
7	13	62
8	14	59.89
9	16	72.28
10	17	83
11	18	65
12	19	78
13	20	85
14		

$M_1 := csort(M_1, 0)$ $X := M_1^{\langle 0 \rangle}$ $Y := M_1^{\langle 1 \rangle}$

Spline coefficients:

$S_1 := lspline(X, Y)$

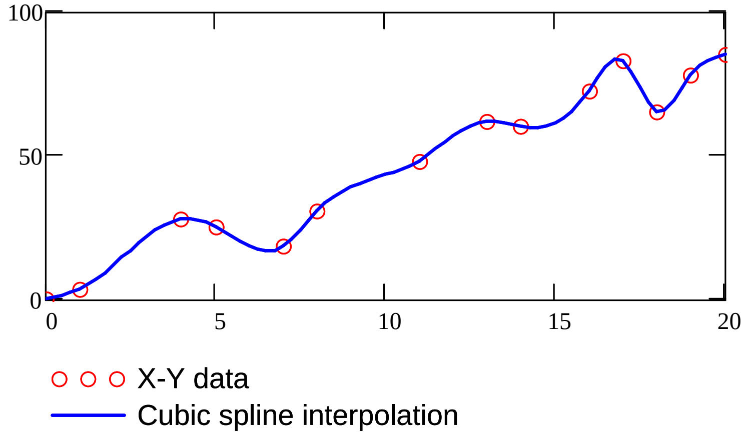
Fitting function:

$fit(x) := interp(S_1, X, Y, x)$

Sample interpolated values:

$z := 0, 0.25 .. 20$

$fit(3) = 21.641$ $fit(9.5) = 41.231$



Any ot the fiiting functions can then be used for calculations (integrals, derivatives etc ...)

Multivariate Regression using *regress*

Assume you have the following data. In this example, X and Y are the independent variables, and Z is the dependent variable.

$i := 0 .. 14$

$X_i :=$ $Y_i :=$ $Z_i :=$

X_i	Y_i	Z_i
47	20	49
80	28	118
125	48	145
31	8	20
116	35	81
99	40	68
43	13	51
30	2	13
150	24	80
112	20	43
42	6	27
150	47	71
92	29	101
89	23	46
46	15	27

The higher dimensional form of regress takes three arguments:

a real array, M_2, each column of which represents data corresponding to one of the independent variables
a real vector, V, representing the data for the dependent variable
a positive integer, n, specifying the degree of the polynomial function to which the data will be fit

$M_2 := \text{augment}(X, Y)$ Combine the independent variables into a single matrix using the *augment* function

$n := 2$ Define the degree of the polynomial. (This value could also be passed directly to *regress*.)

Define the regress function, and view the results.

$$S_2 := \text{regress}\left(M_2, Z, n\right)$$

$$S_2 = \begin{pmatrix} 3.000 \\ 3.000 \\ 2.000 \\ -0.056 \\ 0.092 \\ 2.530 \\ -14.760 \\ 0.691 \\ 2.980 \times 10^{-3} \end{pmatrix}$$

Defining a function using *interp*

You can use the *interp* function to define a two-variable polynomial function which uses these coefficients. *interp* is able to interpret the order of the coefficients and the monomial terms with which they are associated. In particular, it uses the first three terms of the vector, S_2, to determine the built-in function being called (3 for *regress*), the vector position of the first coefficient (always 3), and the degree of the fitting function (n).

$$f(x, y) := \text{interp}\left[S_2, M_2, Z, \begin{pmatrix} x \\ y \end{pmatrix}\right]$$

This definition will vary depending upon the number of independent variables. In this example, we have two independent variables, x and y.

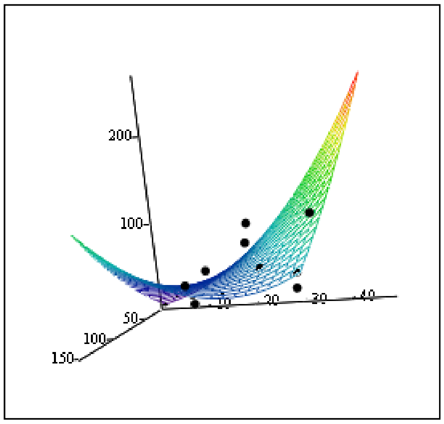
Call the function for specific values but do not use the function to extrapolate. It should be used for interpolation only.

$f(90, 12) = 54.958$

This function call will vary depending upon the number of independent variables you have.

Here we are dealing with a 2D case of multivariate regression. Therefore, we can graph the result as a surface over ranges for x and y.

$F := \mathrm{CreateMesh}(f, \min(X), \max(X), \min(Y), \max(Y), 20, 20)$



The regression surface passes through the space minimizing the distance to each of the data points, shown in black on the graph.

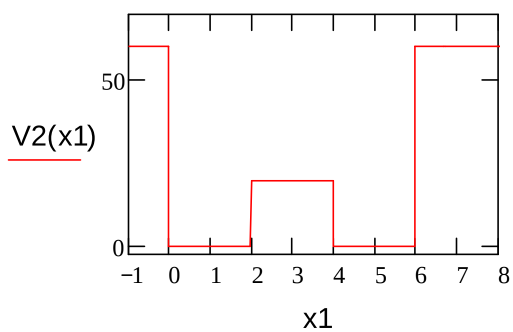
$F, (X, Y, Z)$

Quantum Mechanics

Construction of the double well potential V2(x) and calculation of the ten first energy levels of the system. Plot of the wavefunctions and probability densities. Tunneling effect.

$$V2(x1) := \begin{cases} 0 & \text{if } 0 < x1 < 2 \\ 20 & \text{if } 2 \le x1 < 4 \\ 0 & \text{if } 4 \le x1 < 6 \\ 60 & \text{otherwise} \end{cases}$$

$x1 := -1, -0.999 .. 10$



Differential Equation to solve

$$H := \frac{-h^2}{8\pi^2 \cdot m} \cdot \frac{d^2}{dx^2}$$

$eV := e \cdot volt$ $\qquad eV \equiv 1.602177 \cdot 10^{-19} \cdot J$

$n_1 := 0 .. 9 \qquad h := 6.626 \cdot 10^{-34} \cdot J \cdot s \qquad a := 5 \cdot 10^{-9} \cdot m \qquad m := 9.1 \cdot 10^{-31} \cdot kg$

$$E1(n_1) := \frac{h^2 \cdot {n_1}^2}{8 \cdot m \cdot a^2}$$

$E1(n_1) =$

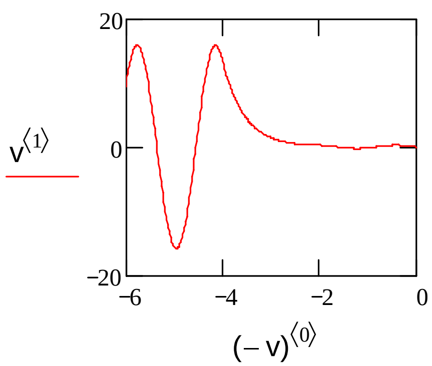
0.000
0.015
0.060
0.136
0.241
0.376
0.542
0.738
0.964
1.220

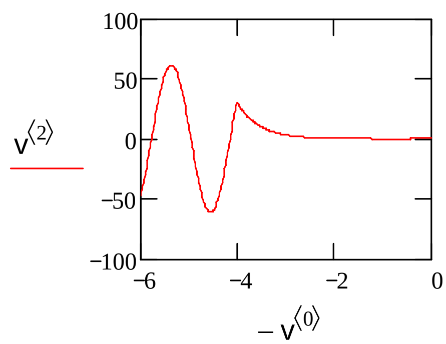
eV

$\varepsilon 1 := 15$ change the values of ε until to get tunneling about 18

$$y1 := \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad D1(x1, y1) := \begin{bmatrix} y1_1 \\ (V2(x1) - \varepsilon 1) \cdot y1_0 \end{bmatrix}$$

$v := \mathrm{rkfixed}(y1, 0, 6, 400, D1)$





No tunneling is observed for ε less than 15. the particle tunnels through the potential barrier at ε greater than 16. At ε greater than 16, tunneling takes place and a wavefunction appears after x = -2.