mathcad data analysis extension pack

UTILITIES

Maxima and Minima

localmax(x, y, [w]) **localmax(x, y, M, [w])**

localmin(x, y, [w]) **localmin(x, y, M, [w])**

The routines **localmax** and **localmin** compare adjacent values in a data set to determine local maxima and minima. The optional window value, w, determines how many points on each side of the point in question must also be smaller or larger for the point to be considered maximum or minimum. Larger windows will help give true maxima for noisy data sets. These functions work for both 2D and 3D data sets.

Local peaks

It's sometimes useful to know where the local maxima and minima are in a data set before doing subsequent processing. Consider the following noisy data mimicking radar antenna response:

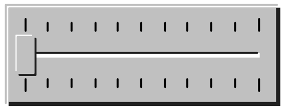
$i := 0..399 \qquad noise := runif(400, -0.1, 0.1)$

$n := 10 \qquad \omega_i := -10 + \frac{20}{399} \cdot i$

Set the data in a two column array

$$Data1^{\langle 0 \rangle} := \omega \qquad Data1^{\langle 1 \rangle} := \frac{n^2}{n + 2\left[\sum_{k=1}^{n-1} [(n-k)\cdot sinc(k\cdot\omega)]\right]} + noise$$

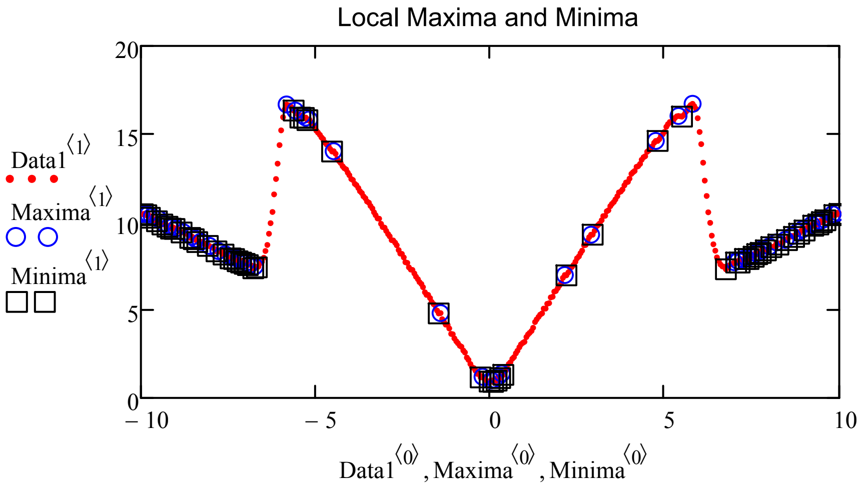
$w :=$



$w := \text{floor}(w) \qquad w = 1$

Set the window width to the place where only real maxima and minima are found.

$\text{Maxima} := \text{localmax}(\text{Data1}, w) \qquad \text{Minima} := \text{localmin}(\text{Data1}, w)$



The graph shows the local maxima and minima using the actual data values. If you wish to know the absolute maximum for approximate interim values, take the numerical derivative of the **interpolated curve** and find the places where it is zero.

Finding functional peaks and troughs

If you know a functional form for your data, perhaps by using **regression** techniques, you can find maxima and minima by searching for the roots of the first derivative:

$$f(x) := 1 - (x+1)^3 + 2\cdot(x+1)^2 + \frac{1}{10}\cdot(x+1)^4$$

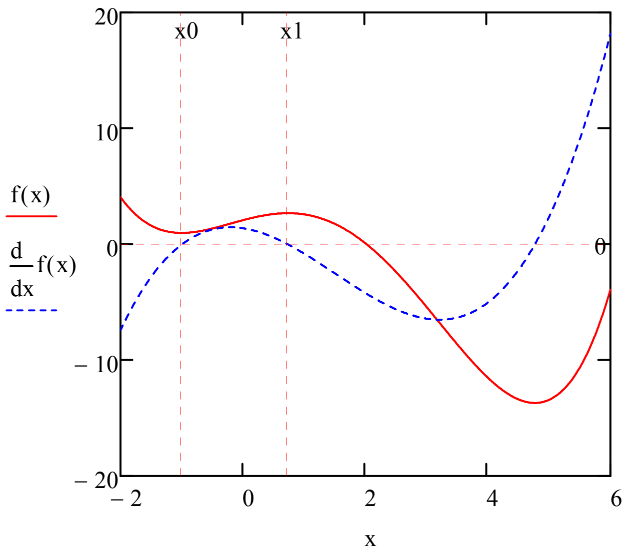
There are numerous ways to do this in Mathcad. The method shown below is applicable to polynomials. First, find the polynomial coefficients using live symbolics:

$$\text{coeff} := \frac{d}{dx}f(x) \;\Bigg|\begin{matrix}\text{coeffs}, x\\ \text{float}, 2\end{matrix} \rightarrow \begin{pmatrix}1.4\\ -0.8\\ -1.8\\ 0.4\end{pmatrix}$$

Use the vector of coefficients as an argument to the polyroots function:

$$zeros := \text{polyroots}(coeff) \qquad zeros = \begin{pmatrix} -1 \\ 0.734 \\ 4.766 \end{pmatrix}$$

$$x0 := zeros_0 \qquad x1 := zeros_1$$



Bump points on the graph of the function f(x) are zero crossings for the derivative, which are found by polyroots. You could use the root function for any functional form to find roots one at a time, or you can try the following program, based on the symbolic processor's root finder:

$$Roots := \frac{d}{dx}f(x) \text{ solve}, x \rightarrow \begin{pmatrix} -1 \\ \frac{\sqrt{65}}{4} + \frac{11}{4} \\ \frac{11}{4} - \frac{\sqrt{65}}{4} \end{pmatrix}$$

$$CriticalPoints := \left| \begin{array}{l} \text{for } i \in 0..\text{last}(Roots) \\ \quad \left| \begin{array}{l} out_{i,0} \leftarrow Roots_i \\ out_{i,1} \leftarrow f(Roots_i) \\ x \leftarrow Roots_i \\ MinOrMax \leftarrow \text{signum}\left(\frac{d^2}{dx^2} f(x)\right) \\ out_{i,2} \leftarrow \text{"Maximum"} \quad \text{if } MinOrMax = -1 \\ out_{i,2} \leftarrow \text{"Minimum"} \quad \text{if } MinOrMax = 1 \\ out_{i,2} \leftarrow \text{"Inflection"} \quad \text{otherwise} \end{array} \right. \\ out \end{array} \right.$$

$$CriticalPoints = \begin{pmatrix} -1 & 1 & \text{"Minimum"} \\ 4.766 & -13.673 & \text{"Minimum"} \\ 0.734 & 2.704 & \text{"Maximum"} \end{pmatrix} \qquad zeros = \begin{pmatrix} -1 \\ 0.734 \\ 4.766 \end{pmatrix}$$

Finding local and global maxima and minima in 3D

Using a demonstration function with one local maximum and 4 local minima, we can use the same empirical **localmax** and **localmin** functions for 3D matrices:

$$g(x, y) := -x^2 + 3 \cdot x - 2 \cdot y^2 - \frac{3}{4} \cdot y + 100$$

$$j := 0..40 \qquad i := 0..40$$

$$x_i := -10 + \frac{i}{2} \qquad y_i := x_i \qquad G_{i,j} := g(x_i, y_j)$$

$$\text{localmin}(x,y,G) = \begin{pmatrix} -10 & -10 & -222.5 \\ -10 & 10 & -162.5 \\ 10 & -10 & -237.5 \\ 10 & 10 & -177.5 \end{pmatrix} \qquad \min(G) = -237.5$$

$$\text{localmax}(x,y,G) = \begin{pmatrix} 0 & 1.5 & 102.25 \end{pmatrix} \qquad \max(G) = 102.25$$

If there is a known analytic function, you can use the Maximize or Minimize functions to find maxima or minima analytically. First, provide guess values to give the solver a place to start seeking a solution.

$$\begin{pmatrix} x \\ y \end{pmatrix} := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

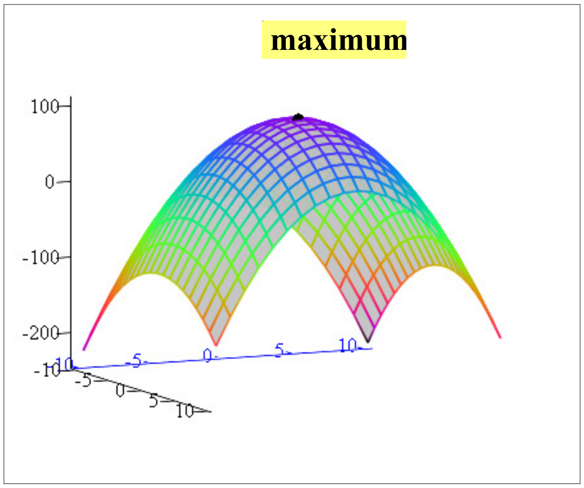
Pass the name of the objective function and the guess values to Maximize.

$$\begin{pmatrix} x_{max} \\ y_{max} \end{pmatrix} := \text{Maximize}(g,x,y) \qquad x_{max} = 1.5 \qquad y_{max} = -0.188$$

When optimizing a function of more than one parameter, Mathcad will return a vector of results. The first element in this vector corresponds to the first variable after the function name in the call to Maximize or Minimize (x in this case), and so on.

$$g(x_{max}, y_{max}) = 102.32$$

$$X_0 := x_{max} \qquad Y_0 := y_{max} \qquad Z_0 := g(x_{max}, y_{max})$$



g, (X, Y, Z)

You can also find maxima and minima in a constrained way using solve blocks. Provide guess values for any unknowns:

$x := 2$ $y := 2$

Given — Open the solve block with Given

$0 \le x \le 5$

$0 \le y \le 5$

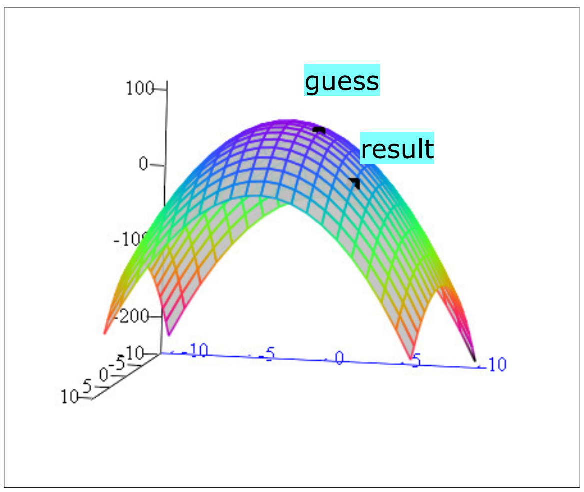
Provide Boolean constraints. These say we want the minimum value for g in the space where x and y can be no greater than 5, nor less than 0.

$$\begin{pmatrix} x_{min} \\ y_{min} \end{pmatrix} := \text{Minimize}(g, x, y)$$

Close with a call to **Minimize**

$X_0 := x$ $Y_0 := y$ $Z_0 := g(x, y)$

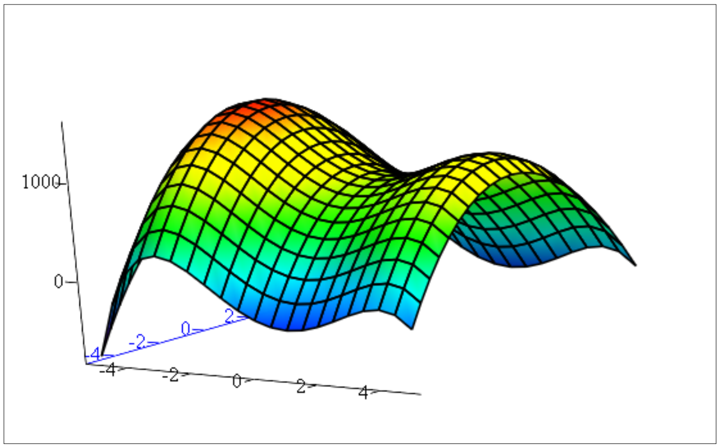
$X_1 := x_{min}$ $Y_1 := y_{min}$ $Z_1 := g(x_{min}, y_{min})$



g, (X, Y, Z)

When optimizing a function in Mathcad, guess values play an extremely important role in the solution which is found. Guesses far from the desired values may cause the solver to converge to a local maximum or minimum. For example, consider the function H, which has two peaks:

$$H(a,b) := 1000 - 3a^4 + 8a^3 + 66a^2 - 144a - 50b^2$$



H

If you compare the results of Maximize with two different sets of guesses, you can see that one set of guesses finds the global maximum, while the other converges to the local maximum.

$a := 4.5$ $\qquad b := 3.5$

$$\begin{pmatrix} a_{max} \\ b_{max} \end{pmatrix} := \text{Maximize}(H, a, b)$$

local max

$a_{max} = 4 \qquad b_{max} = -2.714 \times 10^{-9}$

$H(a_{max}, b_{max}) = 1.224 \times 10^3$

$X := \begin{pmatrix} a \\ a_{max} \end{pmatrix} \qquad Y := \begin{pmatrix} b \\ b_{max} \end{pmatrix}$

$Z := \begin{pmatrix} H(a,b) \\ H(a_{max}, b_{max}) \end{pmatrix}$

$a1 := 1$ $\qquad b1 := -4$

$$\begin{pmatrix} a1_{max} \\ b1_{max} \end{pmatrix} := \text{Maximize}(H, a1, b1)$$

global max

$a1_{max} = -3 \qquad b1_{max} = 2.232 \times 10^{-6}$

$H(a1_{max}, b1_{max}) = 1.567 \times 10^3$

$X1 := \begin{pmatrix} a1 \\ a1_{max} \end{pmatrix} \qquad Y1 := \begin{pmatrix} b1 \\ b1_{max} \end{pmatrix}$

$Z1 := \begin{pmatrix} H(a1,b1) \\ H(a1_{max}, b1_{max}) \end{pmatrix}$