What kind of problem can this example be used to solve?

Suppose you want to plot an equation that has 2 variables (they might be called x and y) and, if the equation is described as LeftSide = RightSide, neither LeftSide nor RightSide is 0.

For example, sin(2*x)*cos(y) = 0.4

Then write this equation with either LeftSide or RightSide subtracted from both sides so that the equation looks like LeftSide-RightSide=0 (or RightSide-LeftSide=0)

In the above example, you could subtract 0.4 to get: sin(2*x)*cos(y)-0.4= 0

Then define a function which equals (in this example) LeftSide-RightSide. When this function = 0, your original equation is true.
The equation needs to be written "= 0" because the implicit-function grapher is designed to plot the points where some expression (the function given by you) = 0.

If there was originally a function z=Funct(x,y), such as z=sin(2*x)*cos(y) in the above example, then graphing Funct(x,y) = K, which is Funct(x,y)-K = 0, where K is a constant, makes part of a contour plot of Funct.

This example is based on http://twt.mpei.ac.ru/ochkov/Lace/Lace_eng.htm

Definitions:

The function to be plotted:

$$f(x, y) := \sin(2\cdot x)\cdot\cos(y) - 0.4$$

Specify what region of the input variables you want to plot (generically, we'll call these variables x and y):

$xmin := -3$ $xmax := 3$ $ymin := -3$ $ymax := 3$

Specify the fineness of display resolution of the plot:

$xstep := 0.02$ $ystep := 0.02$

How big is the graphing problem?

Based on the above constants, let's find out how many points will be potentially graphed:

$$xsize := \frac{(xmax - xmin)}{xstep} + 1 \qquad xsize = 301$$

The graph region is this many points wide

$$ysize := \frac{(ymax - ymin)}{ystep} + 1 \qquad ysize = 301$$

The graph region is this many points tall

$$xsize\cdot ysize = 90601$$

Total number of points potentially graphed

(Scroll down for the implicit-function graphing routine itself)

The implicit-function graphing routine:

It works by computing the value of the function at and near each one of the xsize*ysize points, scanning across the specified region (of the input variables) in grid fashion.

The value ("Pvalue") at each point P is multiplied by the value ("Qvalue") at a point Q that is close to point P (distance "xstep" away from P).
- If P or Q is a root of the equation, then Pvalue or Qvalue is 0, so Pvalue * Qvalue = 0.
- If a root is between P and Q, then, typically, one of Pvalue or Qvalue is positive and the other is negative, so Pvalue * Qvalue < 0. This requires that:
  (a) xstep is small enough so that no other roots are between P and Q, and
  (b) there does not happen to be a root between P and Q that is also a local maximum or minimum [example: if f(x)=x^2 and P is at x=-1 and Q is at x=+1, then Pvalue and Qvalue are both positive, even though there is a root x=0 between P and Q].

So if Pvalue * Qvalue <= 0, the coordinates (x,y) of P are added (concatenated) to a list of points to plot.

The same procedure that has just been described for points P and Q is also done for points P and R, where R is close in the y-direction (by distance ystep) to point P. If only P and Q were considered, then nearly horizontal parts of the graph might not have as many points, which would make them harder to see.

The names P, Q, R are used only in this text, not in the graphing routine itself.
P is (x, y)
Q is (x-xstep, y)
R is (x, y-ystep)

$Points := \begin{pmatrix} 0 & 0 \end{pmatrix}$

Initialize the list of points to plot; otherwise, when the first point is to be added, the software won't be able to detect the dimensions of the list and will indicate an error.

```
for y ∈ ymin , ymin + ystep .. ymax
  for x ∈ xmin , xmin + xstep .. xmax
    if ((f(x - xstep, y)·f(x, y) ≤ 0) ∨ (f(x, y)·f(x, y - ystep) ≤ 0))
      Points := stack(Points, (x y))
    else
      1
```

A dummy "else" clause is used because it was not evident how to specify an "if" with no "else."

$Points := submatrix(Points, 2, rows(Points), 1, 2)$

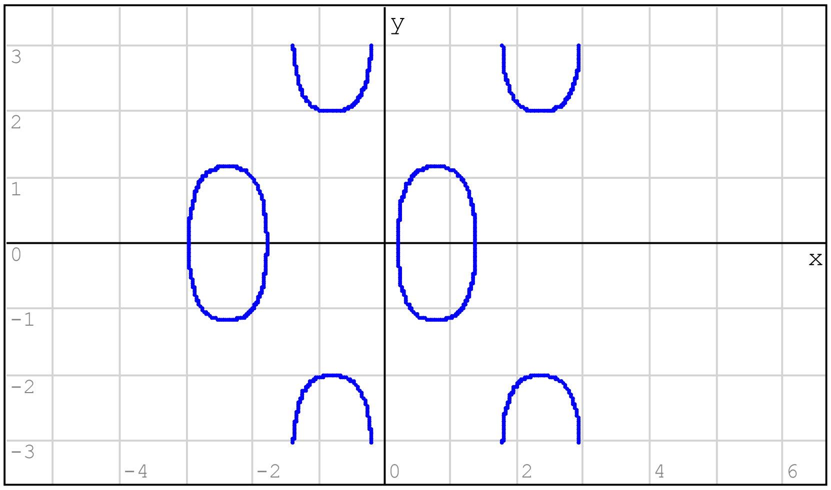
Remove the first "point" that was added when the list of points was initialized. This will NOT erase the point (0, 0), if it happened to be found as a root of the equation or found to be near a root - any (0, 0) that was found will still appear in the list.

(Scroll down for the results)

Results:

rows(Points) = 1188

The number of points that were found to be roots, or to have roots near them.



Points