The results of the Optimize operation are stored in variables. Note that the report that Optimize prints in the history includes only six digits of the values. You can print the results to greater precision using the printf operation and the variables:

```
Printf "The max is at %.15g. The Y value there is %.15g\r", V_maxloc, V_max
```

yields this in the history:

```
The max is at 0.499999480759779. The Y value there is 0.99999999999982
```

Extrema of Multidimensional Nonlinear Functions

Finding extreme points of multidimensional nonlinear functions works very similarly to finding extreme points of a 1D nonlinear function. You provide a user-defined function having almost the same format as for 1D functions. A 2D function will look like this:

```
Function myFunc1(w, x1, x2)
    Wave w
    Variable x1, x2

    return f1(x1, x2)        // an expression...
End
```

This function looks just like the 1D function mySinc we wrote in the previous section, but it has two input X variables, one for each dimension.

We will make a 2D function based on the sinc function. Copy this code into your procedure window:

```
Function Sinc2D(w, xx, yy)
    Wave w
    Variable xx,yy

    return w[0]*sinc(xx/w[1])*sinc(yy/w[2])
End
```

Before starting, let's make a contour plot to see what we're up against. Here are some commands to make a convenient one:

```
Make/D/O params2D={1,3,2}        // nice set of parameters
Make/D/O/N=(50,50) Sinc2DWave    // matrix wave for contouring
SetScale/I x -20,20,Sinc2DWave   // nice range of X values for these functions
SetScale/I y -20,20,Sinc2DWave   // and Y values
Sinc2DWave = Sinc2D(params2D, x, y) // fill f1Wave with values from f1(x,y)
Display /W=(5,42,399,396)        // graph window for contour plot
AppendMatrixContour Sinc2DWave
ModifyContour Sinc2DWave labels=0// suppress contour labels to reduce clutter
```