EE447: Assignment 1 on Minimization
Finding Minima of functions in 1-D and Downhill Symplex method in 2-D

Harishankar Ramachandran
EE Dept, IIT Madras

September 22, 2011

1 Reading Assignment

The minimization chapter in Numerical Recipes.

2 Minimization in 1-Dimension

We wish to find the first ten roots and minima of $J_0(x)$.

1. Bracket the roots and minima by graphing $J_0(x)$. It is always good to know something about these functions. In the case of $J_0(x)$ we have

$$J_\alpha(x) \approx \sqrt{\frac{2}{\pi x}}\cos\left(x - \frac{\alpha\pi}{2} - \frac{\pi}{4}\right)$$

   So try using this to find bracketing points.

2. Find roots and minima to within 10^{-4} by simple minded meshing.

3. Use bisection and Brent's method to locate the roots.

4. Use golden section and Brent's method to locate the minima.

Note: $J_0'(x) = J_1(x)$. So see how accurately you can locate the minimum of $J_1(x)$. That gives an alternative, *more accurate* means to determine the minima of $J_0'(x)$.

How do the number of function evaluations compare?

3 Minimization in n-Dimensions

The function to be minimized is the following:

$$f(x,y) = u^2 + v^2 \tag{1}$$

where

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} 5 \\ 5 \end{pmatrix} \tag{2}$$

where $\alpha = 5\left(0.1\sqrt{x^2+y^2} - 0.5\right)$.

Note: You may implement these routines either entirely in *Python*, as *C* programs run from *Python*, or as *C functions* called from *Python*. I myself chose the second route (though I worked in *Fortran*).

Note: The desired graphs should be overlaid on top of the contour plot obtained in part 1.

Note: Animation means that having got the intermediate locations reached by the solver, you draw the corresponding simplex triangle with timed delays so that we can see how the solver actually crawls its way to the minimum.

1. Implement the function in *Python* and plot the contours. Study the structure of the minima. Notice that the function has a minimum that has to be reached by crawling down valleys that go round and round.

2. Analytically derive the gradient of the function, $f(x,y,z)$. Create a vector plot showing the gradient field and overlay it on the contour plot. Just as the Electric lines of force are normal the equipotentials, the gradient field is normal to the contours you obtained earlier.

3. Create a solver based on the downhill simplex method (following numerical recipes). The code should accept a starting value as an argument. *Save the polygon information at each step*. Plot the way the polygon evolves as it moves towards the solution, for function f.

Study the effect of varying the initial guess on the method's stability and convergence rate.

Note: One thing I noticed was that you have to be very careful with your function definition. I used floats for my function and found that the simplex method had trouble. I think the functions have to be computed in double precision, even if the answers are returned as floats.