Overview of Extensions

- Maximizing
- Global Minima/Maxima
- Beyond 3D

Concept Extension 1: Maximizing

In the past two lessons, we focused on minimizing because that is the most common requirement for an optimization problem.

As you know, one way to convert a minimization procedure to a maximization procedure is to minimize the negative of the function. But if you'll be maximizing frequently, it's nice to have a decicated program for that purpose.

Practice Problem 1

Examine either your steepest-descent program or your conjugate gradient program. What two simple changes would convert it into a maximization program?

Concept Extension 2: Globals

As with most optimization procedures, calculus-based optimizers tend to fall towards the closest minimum or maximum, whether it is local or global.

In unit 2, you wrote a program that tested a grid of points in a function f(x, y) to find the "best" one to use as a seed point for beginning the procedure. This is still a good way to start.

Practice Problem 2

Dig up your grid-testing program from unit 2 and use it to screen points prior to minimizing our problem function

$$f(x, y) = (x - 2)^4 + (x - 2y)^2.$$

(Did that help?)

Concept Extension 3: Beyond 3D

As usual, these lessons have focused on 2-variable, 3-dimensional functions because of the possibility of visualization.

Also as usual, none of the methods used are restricted to 2 variables: gradients, Hessians, directional vectors, and coordinate points can all be made into longer lists or larger matrices, without any need to change the underlying procedures.

Practice Problem 3

Practice Problem 3

Modify your Conjugate Gradient program for a function $f(x, y, z)$ in three variables.

a. Use it to minimize

$$f(x, y, z) = 3(x + z - 2)^2 + (y - 2z)^2 + 2x^2$$

starting at (5, 5, 5).

b. Check your work by finding the gradient *by hand*, setting it equal to 0 and solving to find the critical point, then calculating the Hessian by hand and its eigenvalues in Julia to verify your critical point is a minimum.