Assignment 5.

Given March 7, due February 21.

Objective: Work with optimization methods and robust software.

This assignment asks you to write good, robust numerical software for finding the minimum of a function of several variables. Pay particular attention to independent testing, modularity, and graceful failure. Make sure the test cases for individual components sometimes lead to failure so you can see the failure mechanisms working. In general, it is important to design testers to exercise all features of the code.

1. Line Search: Write a program that attempts to minimize a function of a single variable, $\phi(t)$. The algorithm should have two phases. The first phase assumes that $\phi'(0) < 0$ and searches for a positive t that is within a factor of 2 of a local minimum. Clearly this will be the case if

$$\phi(t) < \phi(0) \quad \text{and} \quad \phi(t) < \phi(2t) \ . \tag{1}$$

A binary search algorithm for locating a suitable t would start with an initial guess, say $t = 1$. Evaluate $\phi(t)$. If $\phi(t) > \phi(0)$, then replace t by $t/2$. If $\phi(t) < \phi(0)$, evaluate $\phi(2t)$. If $\phi(2t) < \phi(t)$, replace t with $2t$. Repeat until t satisfies (1) or until an iteration bound is exceeded. For example, if $\phi(t) = -t$ then a binary search strategy will not find a local minimum.

The second phase is a Newton search for the minimum. This should have a safeguard against negative ϕ'' and a safeguard against wild iterates (e.g., ones that violate the criteria of phase 1). Try your algorithm on the function $\phi(t) = e^t - 1/(1+t^2)$ with various initial guesses. Large positive or negative initial guesses present challenges. Observe the action of phase one and the eventual quadratic convergence under phase two. Observe the iterates of unsafeguarded Newton's method from a large positive initial guess. We will only use phase one for the line search part of multidimensional optimization below.

2. Write a program that performs unsafeguarded Newton's method for a function of several variables. Use your Choleski factorization routine from assignment 4. Make sure to test whether the Choleski factorization succeeded and fail gracefully if it did not. Try your program on the function

$$f(x,y) = \phi\left(x^2 + a(y - b\sin(cx))^2\right) \tag{2}$$

with $a = 2$, $b = 1$, and $c = 1$. Here, ϕ is the function of t from part 1. Try initial guesses close and far from the solution ($x = y = 0$). With a good initial guess you should see quadratic convergence to the minimizer. With a poor initial guess you should see wild behavior or termination with failure.

3. Add two safeguards to the program from step 2. Do a phase one line search to make sure the step is not crazy and check that the Newton step is a descent direction. Use the negative gradient as a search direction if not[1] Use this to minimize the objective function (1) with $a = 30$, $b = 1$, and $c = 6$, starting from initial guess $(x_1, x_2) = (3, 1)$. Make a contour plot of the objective function showing the minimization iterates.

[1] The "modified Choleski" decomposition is much better than the identity Hessian (negative gradient) search direction. I suggest the worse strategy here because it will not force you to revisit your Choleski factorization routines.