CS205 - Class 7

Readings: Heath 5.6

Systems of Nonlinear Equations

1. Let's turn our attention back to systems of nonlinear equations, i.e. A(x)=b or F(x)=0.
   a. Here the **Jacobian** matrix, J(x), is rather useful as a linearization of the nonlinear problem.
   b. Here we define $J_{ij} = \partial F_i / \partial x_j$ where each equation of F(x)=0 is written individually as $F_i(x) = 0$, and each x_j is the j-th component of the x vector.
   c. For example, consider $F_1(x) = x_1 + \sin x_2 + 4 = 0$ and $F_2(x) = (x_1)^2 + x_2 = 0$ which can be written in matrix form as $F(x) = \begin{pmatrix} x_1 + \sin x_2 + 4 \\ (x_1)^2 + x_2 \end{pmatrix} = 0$. Then the Jacobian matrix is $J(x) = \begin{pmatrix} 1 & \cos x_2 \\ 2x_1 & 1 \end{pmatrix}$.
   d. Note that J(x), that is J, generally depends on the x vector.
   e. In general, we write J(x)=F'(x) and note that the Jacobian is the multidimensional generalization of f'(x).
   f. Thus conditioning in multiple dimensions depends on the Jacobian matrix just as conditioning for scalars depends on f'(x).
   g. Moreover, we desire a nonsingular J, at least "locally" to the solution, in order to proceed.
2. Newton's method for F(x)=0 is $x_{k+1} = x_k - J^{-1}(x_k) f(x_k)$.
   a. Note that $1/f'(x_k)$ is replaced by $J^{-1}(x_k)$.
   b. Instead of computing the inverse, we solve the linear system $J(x_k) y_k = -f(x_k)$ where $y_k = x_{k+1} - x_k$.
      i. That is, we first solve the linear system to find the increment y_k and then calculate $x_{k+1} = x_k + y_k$
      ii. Now one can see why J needs to be nonsingular for x_k "close" to the solution x^*, i.e. because we need to solve the linear system
   c. This can be very computationally expensive since one has to repeatedly solve a linear system.
   d. Also, one might worry about robustness in case *any* one of the linear system solvers fails
   e. *More times than not, one has to repeatedly solve linearly systems, i.e. it's not a one time deal. That's precisely why it is important to look for symmetry, sparsity, and structure in problem formulations.*
   f. There are secant-like methods in multiple dimensions that compute the Jacobian matrix in pieces "on the fly".
      i. Broyden's Method
         1. start with $J_0 = I$
         2. solve $J_k y_k = -f(x_k)$ and set $x_{k+1} = x_k + y_k$ as usual
         3. $J_{k+1} = J_k + \frac{(F(x_{k+1}) - F(x_k) - J_k y_k) y_k^T}{y_k \cdot y_k}$