Math 451 – DeLillo
Note on Newton's method for systems of nonlinear equations – 9/19/12

Newton's method for systems of nonlinear equations, like Newton's method for solving $f(x) = 0$ for a single-valued nonlinear function f of one variable x, is a based on producing a sequence of solutions $x^{(k)}$ to successive linear problems such that $x^{(k)}$ converges to the solution x_{exact}. The linear problems are derived from linear approximations to $f(x)$ at $x = x^{(k)}$. Suppose we have a (nonlinear) vector-valued function $F(X)$ of n variables $X = [x_1, x_2, \ldots, x_n]^T]$ where F has n components $f_j(X), j = 1, 2, , \ldots, n$, and we wish to find a zero X_{exact} of F, such that $F(X_{exact}) = [0, 0, \ldots, 0]^T$. That is, we wish to solve

$$F(X) = \begin{bmatrix} f_1(x_1, x_2, \ldots, x_n) \\ f_2(x_1, x_2, \ldots, x_n) \\ \vdots \\ f_n(x_1, x_2, \ldots, x_n) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Since there are n equations in n unknowns, there is some hope that we may have a solution. In fact, usually there are multiple solutions (or none), just as in the scalar case. Recall from Calculus III (where usually $n = 2$ or 3) that scalar-valued functions $f_j(x_1, x_2, \ldots, x_n)$ can be approximated by a linear function (the tangent plane for $n = 2$) at a given point $X = X^k = [x_1^k, x_2^k, \ldots, x_n^k]^T$, ($x_i^k$ here denotes the kth iterate of x_i and NOT x_i raised to the kth power) that is by

$$\begin{aligned} f_j(x_1, x_2, \ldots, x_n) \quad &\approx \quad f_j(x_1^k, x_2^k, \ldots, x_n^k) + \frac{\partial f_j(x_1^k, x_2^k, \ldots, x_n^k)}{\partial x_1}(x_1 - x_1^k) \\ &+ \quad \frac{\partial f_j(x_1^k, x_2^k, \ldots, x_n^k)}{\partial x_2}(x_2 - x_2^k) + \cdots \\ &+ \quad \frac{\partial f_j(x_1^k, x_2^k, \ldots, x_n^k)}{\partial x_n}(x_n - x_n^k), \end{aligned}$$

for $j = 1, 2, \ldots, n$. We can write this in matrix-vector notation as

$$F(X) \approx F(X^k) + J(X^k)(X - X^k),$$

where $J(X^k)$ is the $n \times n$ Jacobian matrix,

$$J(X) = \begin{bmatrix} \frac{\partial f_1(X)}{\partial x_1} & \frac{\partial f_1(X)}{\partial x_2} & \dots & \frac{\partial f_1(X)}{\partial x_n} \\ \frac{\partial f_2(X)}{\partial x_1} & \frac{\partial f_2(X)}{\partial x_2} & \dots & \frac{\partial f_2(X)}{\partial x_n} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial f_n(X)}{\partial x_1} & \frac{\partial f_n(X)}{\partial x_2} & \dots & \frac{\partial f_n(X)}{\partial x_n} \end{bmatrix}$$

evaluated at $X = X^k$. For the Newton step, we solve the linear system,

$$JU^k = -F(X^k)$$

for the Newton update $U^k = X - X^k$, (an $n \times 1$ vector). We can do this easily in MATLAB using the Gaussian elimination with the backslash \,

$$U^k = -J\backslash F(X^k).$$

Then we form the update

$$X^{k+1} = X^k + U^k = X^k - J\backslash F(X^k)$$

and repeat the process until the components of either $F(X^k)$ or U^k get sufficiently small, just as in the scalar case.

Here is a simple example from the class diary. Let

$$F = \begin{bmatrix} x_1^2 + x_2^2 - 1 \\ x_1^2 - x_2^2 \end{bmatrix}.$$

The solutions to $F = 0$ are obviously any combinations of $x_1 = \pm\sqrt{2}/2, x_2 = \pm\sqrt{2}/2$. The Jacobian matrix is

$$J = \begin{bmatrix} 2x_1 & 2x_2 \\ 2x_1 & -2x_2 \end{bmatrix}.$$

In the diary for 9/19/12 we started with an initial guess $X^1 = [1, 1.3]^T$ and saw that the Newton iterations converged rapidly to the solution $[\sqrt{2}/2, \sqrt{2}/2]$.