Lecture 3
Newton's Method and Loops

Solving equations numerically

For the next few lectures we will focus on the problem of solving an equation:

$$f(x) = 0. \tag{3.1}$$

As you learned in calculus, the final step in many optimization problems is to solve an equation of this form where f is the derivative of a function, F, that you want to maximize or minimize. In real engineering problems the function you wish to optimize can come from a large variety of sources, including formulas, solutions of differential equations, experiments, or simulations.

Newton iterations

We will denote an actual solution of equation (3.1) by x^*. There are three methods which you may have discussed in Calculus: the bisection method, the secant method and Newton's method. All three depend on beginning close (in some sense) to an actual solution x^*.

Recall Newton's method. You should know that the basis for Newton's method is approximation of a function by it linearization at a point, i.e.

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0). \tag{3.2}$$

Since we wish to find x so that $f(x) = 0$, set the left hand side ($f(x)$) of this approximation equal to 0 and solve for x to obtain:

$$x \approx x_0 - \frac{f(x_0)}{f'(x_0)}. \tag{3.3}$$

We begin the method with the initial guess x_0, which we hope is fairly close to x^*. Then we define a sequence of points $\{x_0, x_1, x_2, x_3, \ldots\}$ from the formula:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, \tag{3.4}$$

which comes from (3.3). If $f(x)$ is reasonably well-behaved near x^* and x_0 is close enough to x^*, then it is a fact that the sequence will converge to x^* and will do it very quickly.

The loop: `for ... end`

In order to do Newton's method, we need to repeat the calculation in (3.4) a number of times. This is accomplished in a program using a *loop*, which means a section of a program which is repeated. The simplest way to accomplish this is to count the number of times through. In MATLAB, a `for ... end` statement makes a loop as in the following simple function program: