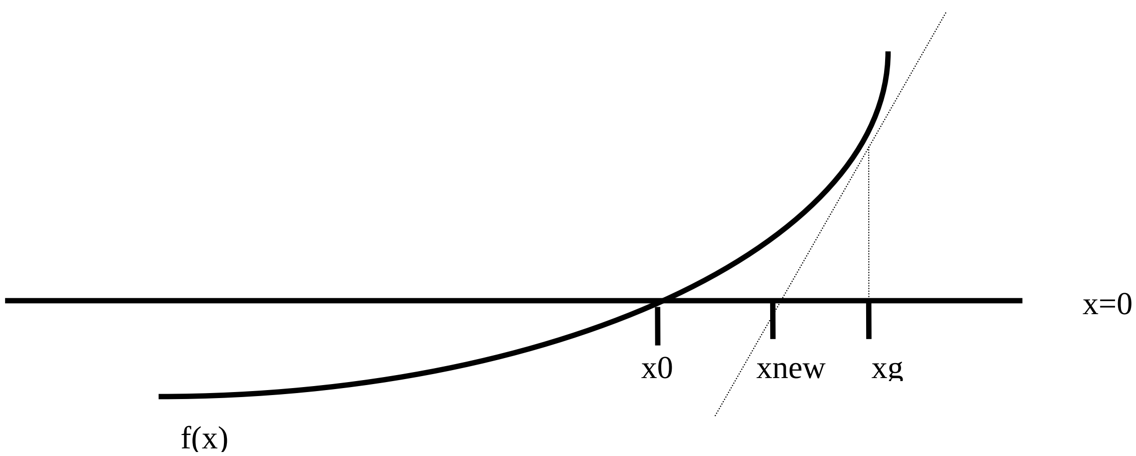
NEEP 602 -- Engineering Problem Solving II
Exercise 12

Elliptic Partial Differential Equations (Speeding Up Convergence)

Here we consider a way to accelerate the convergence of elliptic PDE's. The technique is called successive over-relaxation (SOR). we'll first look at a similar effect in one dimension to demonstrate the technique, and then we'll look at how you implement it for elliptic PDE's in Excel.

A 1-D Analog

Consider Newton's method for finding the roots of functions of one variable. As shown graphically below, this technique starts with a guess, extrapolates a tangent to the curve until it crosses the axis, and then uses this crossing point as a new guess. Note how the iteration underestimates the amount needed to reach the real root.



If we simply increase the correction by a predetermined factor, we can accelerate the convergence. Obviously, for some functions convergence would be harmed by increasing the increment prescribed by Newton's method, but for a large class of elliptic PDE's, the two-dimensional analog of this technique dramatically improves convergence.

Successive Overrelaxation

To demonstrate overrelaxation for elliptic PDE's, we'll consider the example discussed in the previous exercise. That is, we'll deal with heat conduction problems in two dimensions. The governing equation for these problems is:

$$\nabla^2 T + \frac{Q}{k} = 0$$

where T is the temperature, Q is the volumetric heating, and k is the thermal conductivity of the solid. Differencing this equation and using equal mesh spacing in both directions yields:

$$T_{i,j} = \frac{T_{i,j+1} + T_{i,j-1} + T_{i+1,j} + T_{i-1,j}}{4}.$$

Now we must massage this equation to make it suitable for SOR. We do this by adding and subtracting a term from the right side of the equation, yielding

$$T_{i,j} = T_{i,j} + \frac{T_{i,j+1} + T_{i,j-1} + T_{i+1,j} + T_{i-1,j} - 4T_{i,j}}{4}$$

Written in this fashion, the second term on the right side of this equation can be thought of as the correction to the current guess. Next we add an over-relaxation parameter (ω), yielding:

$$T_{i,j} = T_{i,j} + \omega\left(\frac{T_{i,j+1} + T_{i,j-1} + T_{i+1,j} + T_{i-1,j} - 4T_{i,j}}{4}\right)$$

For ω greater than 1, this provides over-relaxation.

In order to yield optimum convergence, we must choose the value for the over-relaxation parameter that minimizes the number of iterations to convergence. For the classes of equations we've been considering, the optimal value is found from

$$\omega = \frac{2}{1 + \sqrt{1 - \rho^2}}$$

$$\rho = \frac{\cos\left(\frac{\pi}{J}\right) + \left(\frac{h}{s}\right)^2 \cos\left(\frac{\pi}{L}\right)}{1 + \left(\frac{h}{s}\right)^2}$$

where h and s are are the mesh spacings in the x and y directions, respectively, J is the number of mesh points in the x direction, and L is the number of points in the y direction.