Non-Linear Scalar Equations

To this point, our ordinary differential equations have been linear and readily solved via $[\text{LHS}]\{Q\} = \{\text{RHS}\}$. However, non-linearities frequently arise in engineering problem statements, i.e. temperature-dependent thermal conductivity, and a solution methodology must be developed to solve our recipe induced matrix statement. We shall begin with a scalar equation to demonstrate the process and then apply it to a non-linear heat conduction problem.

Assume we have a non-linear equation for which we seek the roots

$$y(x) = 5x^2 + 3x - 2 \tag{a}$$

If the equation is simple enough, an algebraic solution may be readily obtainable:

$$5x^2 + 3x - 2 = 0 \quad \Rightarrow \quad x = \frac{-3 \pm \sqrt{3^2 - 4*5*(-2)}}{2*5} = -1,\ \frac{2}{5} \tag{b}$$

Unfortunately, algebraic approaches are not amenable to numerical methods as we seek a general purpose technique for obtaining the roots. An elementary approach is to employ bisection and make a series of refined guesses until the LHS is approximately the RHS. This is a very inefficient approach and will fail if the solution does not fall within the initial bounds.

Iterative methods utilize an initial guess and, based on the difference between the result and the desired result, “correct” the guess until the solution converges to within a desired tolerance. Classic examples include Gauss-Seidel and Successive Over Relaxation (SOR). The methods, while easy to understand and implement, exhibit a *linear* convergence. While this is faster than bisection, it is slower than Newton which exhibits *quadratic* convergence. Heed that these techniques are all employed in FEA/CFD packages and the solution technique is typically an option to be selected by the user.

Newton iteration (our preferred choice) is performed via the following recipe:

1. Our new (current) guess at the “correct” root will be the old (previous) guess plus an incremental change that moves the guessed root towards the correct solution.

$$Q^{p+1} = Q^p + dQ^{p+1} \tag{1}$$

2. The incremental change is obtained from the Jacobian and Residual of the equation evaluated with the old (previous) value of Q

$$JAC^p dQ^{p+1} = -RES^p \tag{2}$$

   where the Jacobian is the derivative of the Residual and the Residual is the original equation for which we seek the root.

$$JAC^p \equiv \left(\frac{\partial RES}{\partial Q}\right)^p \quad \text{and} \quad RES^p = f(Q)^p \tag{3}$$

3. Having evaluated (2) using (3), a new (current) guess for the root Q is obtained with (1).
4. The new (current) guess for Q then becomes the old (previous) value and the process is repeated until the change $\Delta Q^{p+1} = Q^{p+1} - Q^p$ is within some tolerance, i.e. the solution has converged.

Let us apply the Newton Iteration Algorithm to (a) by hand to get a feel for the recipe. We must first identify RES and then perform the calculus to obtain JAC. This step is the main detraction from Newton!

$$RES = 5x^2 + 3x - 2$$
$$JAC = \frac{\partial RES}{\partial x} = \frac{d}{dx}\left(5x^2 + 3x - 2\right) = 10x + 3$$

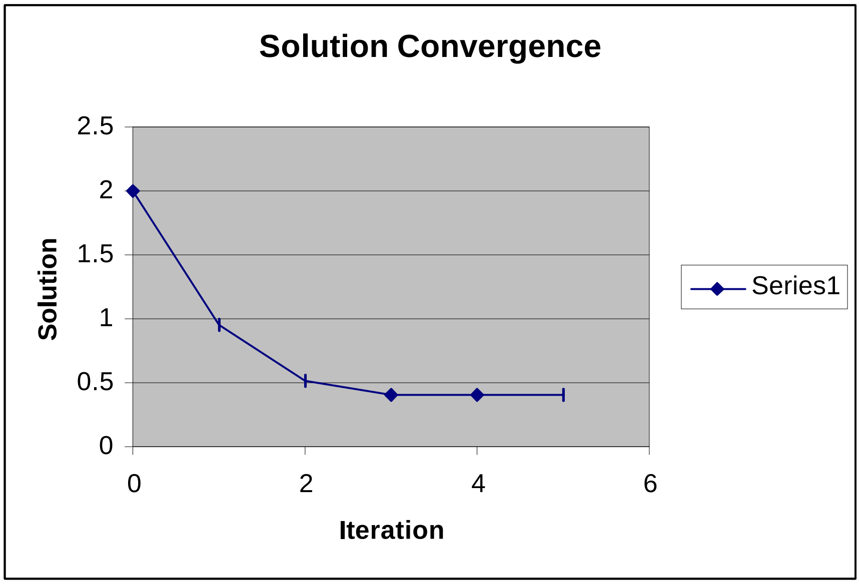
p	Q^p	JAC^p	RES^p	dQ^{p+1}	Q^{p+1}	$\left\|\Delta Q^{p+1}\right\|$
1	2	23	24	$-\frac{24}{23}$	0.9565	1.0435
2	0.9565					
3						
4						
5						

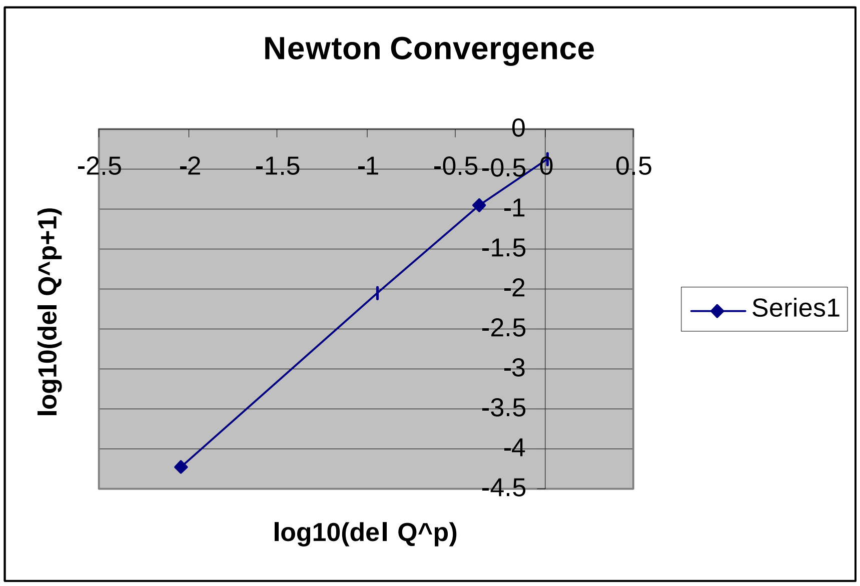
Iteration 1

$$Q^p = 2 \qquad JAC^p = 10 * 2 + 3 = 23 \qquad RES^p = 5 * 2^2 + 3 * 2 - 2 = 24$$

$$dQ^{p+1} = -\frac{RES^p}{JAC^p} = -\frac{24}{23} \qquad Q^{p+1} = Q^p + dQ^{p+1} = 2 - \frac{24}{23} = 0.9565 \qquad \left|\Delta Q^{p+1}\right| = \left|0.9565 - 2\right| = 1.0435$$

Iteration 2


Solution Convergence
Solution
2.5
2
1.5
1
0.5
0
0
2
4
6
Iteration
Series1


Newton Convergence
log10(del Q^p+1)
0
-2.5
-2
-1.5
-1
-0.5
-0.5
0
0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
-4.5
log10(del Q^p)
Series1