A brief note on root finding in nonlinear equations

Consider solving an equation in one variable x

$$f(x) = 0 \tag{1}$$

In this course we will assume $f(x)$ to be a smoothly varying function that can have extrema in the range of our interest. Finding root(s) numerically always start with a guess *i.e.* an approximate trial solution, then proceeds by iteration using an algorithm which improves the solution until some predetermined convergence criterion is meet. In this course, we will always assume that the function $f(x)$ is continuous at the position of the root. For convergence, it is necessary to have a "good" *i.e.* an educated initial guess. This might be achieved by plotting $f(x)$ vs. x to get some idea of the root. In this course we will learn 3 methods for finding roots of nonlinear equations.

1. Bisection method

2. False Position (Regula Falsi) method

3. Newton-Raphson method

Bisection method

The Bisection method starts with *bracketing* meaning choosing two points a and b such that $f(a)$ and $f(b)$ have opposite signs. If the function is continuous then at least one root must lie in the interval. However, the bracketing can go wrong if $f(x)$ has double roots or $f(x) = 0$ is an extrema or $f(x)$ has many roots over a small interval. The steps involve in bracketing are,

1. Choose a and b, where $a < b$, and calculate $f(a)$ and $f(b)$.

2. If $f(a) * f(b) < 0$ then bracketing done.

3. If $f(a) * f(b) > 0$ *i.e.* same sign, then check whether $|f(a)| \lesseqgtr |f(b)|$.

4. If $|f(a)| < |f(b)|$, shift a further to the left by using, say, $a = a - \beta * (b - a)$ and then go back to second step. Choose your own β, say 1.5.

5. If $|f(a)| > |f(b)|$, shift b further to the right by using, say, $b = b + \beta * (b - a)$ and then go back to second step. Choose your own β, say 1.5.

6. Give up if you can't satisfy the condition $f(a) * f(b) < 0$ in $10 - 12$ iterations. Start with a new pair $[a', b']$ and do the thing all over again.

Now the bisection method proceeds as

1. Choose appropriate $[a, b]$, where $a < b$, to bracket the root.

2. Bisect the interval
$$c = \frac{a+b}{2}$$

3. If $f(a)*f(c) < 0$, then root lies to the left of c and replace $b = c$. Done if $|b-a| < \epsilon$, where $\epsilon = 10^{-4}$ say, else iterate the bisection of step 2.

4. If $f(a) * f(c) > 0$, then root lies to the right of c and replace $a = c$. Done if $|b - a| < \epsilon$, where $\epsilon = 10^{-4}$ say, else iterate the bisection of step 2. Alternatively, we can also test $|f(c)| < \epsilon$ for convergence since at root x^* implies $f(x^*) = 0$.

This method has slowest convergence of all other root finding methods but it is a sure shot to root provided you can bracket the root.

False Position method

In False Position method, one does some sort of interpolation to converge o a root faster than Bisection. The method involves finding the slope of the straight line joining $[a, b]$, which bracketed the root and finding where this line crosses the abscissa (x-axis) and take that point as either of a or b depending on the relative sign of the functions at those points. If the function $f(x)$ is convex in the interval $[a, b]$ that contains a root then one of the points a or b is always fixed and the other point varies with iterations. This makes checking for convergence little tricky.

1. Choose appropriate $[a, b]$, where $a < b$, to bracket the root.

2. Calculate the slope of the straight line joining a and b and obtain c where the line crosses the abscissa,
$$m = \frac{f(b) - f(a)}{b - a} = \frac{f(b) - 0}{b - c} \quad \Rightarrow \quad c = b - \frac{(b - a) * f(b)}{f(b) - f(a)}$$

3. If $f(a) * f(c) < 0$, then root lies to the left of c and replace $b = c$. With new b, done if $|b - a| < \epsilon$, else iterate step 2.

4. If $f(a) * f(c) > 0$, then root lies to the right of c and replace $a = c$. With new a, done if $|b - a| < \epsilon$, else iterate step 2. However, this convergence testing will possibly not work if $f(x)$ is convex at the root. Try, instead, $|f(c)|$ or $|c_{i+1} - c_i| < \epsilon$ for convergence.

Newton-Raphson method

The most famous, but not necessarily most efficient, of all root finding methods is Newton-Raphson. This works also for multivariate functions. Unlike the previous two methods, this one involves both $f(x)$ and its derivative $f'(x)$. And finally, it converges quadratically, meaning near a root the number of significant digits approximately doubles with each step. The method is based on Taylor series expansion, to solve $f(x) = 0$ Taylor expand $f(x)$ at any point x_0,

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{1}{2!}(x - x_0)^2 f''(x_0) + \cdots \tag{2}$$

If we are closer to the root, we can stop at $f'(x)$ term,

$$f(x) = f(x_0) + (x - x_0)f'(x_0) = 0 \quad \Rightarrow \quad x = x_0 - \frac{f(x_0)}{f'(x_0)} \tag{3}$$

Far from a root, the higher derivative terms in the series become important. For non-linear $f(x)$, the resulting x found above can be treated as an approximate root, which can be improved iteratively to move from the x_0 towards the root,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \qquad n = 0, 1, 2, \ldots \tag{4}$$

You must have noticed there is no bracketing here, just an initial guess x_0 but involved taking a derivative,

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h} \tag{5}$$

The steps are absolutely straight forward,

1. Make a good guess of x_0

2. Evaluate $f(x)$ and its derivative $f'(x)$ at $x = x_0$.

3. Iterate as given in (4) until $|x_{n+1} - x_N| < \epsilon$.

Root(s) of Polynomials

A polynomial of degree n has n roots $x_i,\ (i = 1, 2, \ldots, n)$,

$$P(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n = (x - x_1)\,(x - x_2)\cdots(x - x_n) \qquad (6)$$

For polynomials with real coefficients a_i the roots can be real or complex, which occur in pairs that are conjugate. In this course we will restrict ourselves to only real roots. Firstly, we will find a root, say x_1, using **Laguerre's method**. Then we go for deflating the polynomial *i.e.* obtain a reduced polynomial $Q(x)$ of degree one less than $P(x)$,

$$P(x) = (x - x_1)\,Q(x) \qquad (7)$$

where the roots of $Q(x)$ are the remaining roots of $P(x)$. Next find a root of $Q(x)$ by the same Laguerre method and deflate it as in (7) to the next lower degree polynomial and so one till the end. The Laguerre algorithm is as follows,

1. Choose an initial guess α_0.

2. If α_0 is bang on one of the roots, go for deflation.

3. Else calculate the following

$$G = \frac{P'(\alpha_i)}{P(\alpha_i)}, \quad H = G^2 - \frac{P''(\alpha_i)}{P(\alpha_i)} \quad \Rightarrow \quad a = \frac{n}{G \pm \sqrt{(n-1)(nH - G^2)}}$$

   Choose the sign in the denominator of a such as to give the denominator the larger absolute value.

4. Set $\alpha_{i+1} = \alpha_i - a$ as new trial.

5. Continue iteration till $|\alpha_{i+1} - \alpha_i| < \epsilon$ and set $x_1 = \alpha_i$.

6. Go for deflation to reduce the degree of the polynomial and do the above iteration all over again to find x_2 etc.

For deflation, we have to divide the polynomial $P(x)$ by $(x - x_1)$, then $Q(x)$ by $(x - x_2)$ and so on. The method used is the regular *synthetic division* method.

1. To divide $P(x)$ by $(x - x_1)$ (the leading coefficient must always be 1), arrange the terms in $P(x)$ in descending order of power and store the coefficients with 0 as the coefficient(s) of the missing power(s). For example,

$$\frac{P(x)}{x - x_1} = \frac{-x^3 + 3x^2 - 4}{x - 2} \quad \Rightarrow \quad \text{divisor} = 2, \quad \text{coefficients} = [-1,\ 3,\ 0,\ -4]$$

2. Bring down the coefficient of the leading power below the horizontal line. Multiply the coefficient of leading power with the divisor and add it to the coefficient of the next lower power and bring it down below the horizontal line again. Continue this process till the end.

$$\begin{array}{c|cccc} & -1 & 3 & 0 & -4 \\ 2 & + & -2 & 2 & 4 \\ \hline & -1 & 1 & 2 & 0 \end{array}$$

If x_1 is a root then the last sum, which gives the remainder, must be zero. Now for the reduced lower degree polynomial with the numbers below the line

$$\frac{P(x)}{x - x_1} = \frac{-x^3 + 3x^2 - 4}{x - 2} = -x^2 + x + 2 = Q(x)$$

3. Repeat the above process with $Q(x)$ and keep doing for successive roots till you get the final monomial $(x - x_n)$.

$$\begin{array}{c|ccc} & -1 & 1 & 2 \\ 2 & + & -2 & -2 \\ \hline & -1 & -1 & 0 \end{array} \quad \Rightarrow \quad Q(x) = -x - 1$$

The polynomial in the example is thus factorized in terms of its roots as $P(x) = -x^3 + 3x^2 - 4 = (x - 2)\,(x - 2)\,(-x - 1)$.