A short note on finding roots of nonlinear equations

Consider solving a nonlinear equation in one variable x

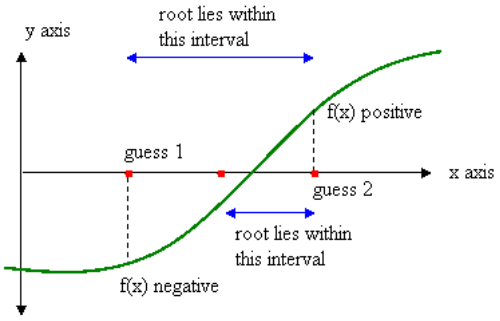
$$f(x) = 0 \tag{1}$$

In this course we will assume $f(x)$ to be a smoothly varying function that can have extrema, minima and/or maxima, over the range we are interested in. A few typical examples of $f(x)$ are

$$f(x) = \cos x - x^3, \quad 3x + \sin x - \exp x, \quad x \exp x - 2, \quad x^3 + 3x - 5 = 0 \tag{2}$$

If a value x_0 in the interval (a_0, b_0) satisfies the equation (1) *i.e.* $f(x_0) = 0$, then x_0 is a *root* or *zero* of the function $f(x)$ and is **one** of the solutions in that interval. Since $f(x)$ is a continuous function and there exist two points a_0 and b_0 such that $f(a_0)$ and $f(b_0)$ are of opposite signs, then according to *intermediate value theorem*, the function $f(x)$ has a root in the interval (a_0, b_0).

Finding root numerically always start with guesses, two x-values a_0, b_0 either found by trial-and-error or educated guess, at which $f(x)$ has opposite signs. Since $f(x)$ is continuous and one root of it is guaranteed to lie between these two values, we say these a_0 and b_0 *bracket* the root. Then we proceed by iterations to produce a sequence of shrinking intervals $(a_0, b_0) \to (a_i, b_i)$ such that the shrunk intervals always contain one root of $f(x)$.



For convergence, it is necessary to a *good* 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 4 methods for finding roots of nonlinear equations including the one specialized for finding the roots of polynomials.

1. Bisection method
2. False position (Regula falsi) method

3. Newton-Raphson method

4. Laguerre's method

Bisection method

The bisection method is the simplest but relatively slow method of finding root of nonlinear equations. The method is guaranteed to converge to a root of $f(x)$ if the function is continuous in the interval $[a, b]$ where $f(a)$ and $f(b)$ have opposite signs. But bracketing can go wrong if $f(x)$ has double roots or $f(x) = 0$ is an extrema or $f(x)$ has many roots over the interval chosen. 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. Proceed to execute bisection method.

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 *i.e.* $f(a) * f(b) < 0$.

2. Bisect the interval, the midpoint of the interval is taken as first approximation with $a_1 = a$ and $b_1 = b$

$$c_1 = \frac{b_1 + a_1}{2} \tag{3}$$

   The maximum absolute error of this approximation is

$$|c_1 - \bar{x}| \leq \frac{b_1 - a_1}{2} = \frac{b - a}{2} \tag{4}$$

3. If the error in (4) is considered too large, repeat the above step with new interval either $[a_2, b_2] = [a_1, c_1]$ or $[c_1, b_1]$ depending on the sign of $f(c_1)$. The new bisection or midpoint is $c_2 = (b_2 + a_2)/2$ and maximum absolute error is

$$|c_2 - \bar{x}| \leq \frac{b_2 - a_2}{2} = \frac{b - a}{4}$$

4. If in the n-th step the corresponding values are a_n, b_n, c_n then

$$c_n = \frac{b_n + a_n}{2} \quad \rightarrow \quad |c_n - \bar{x}| \leq \frac{b_n - a_n}{2} = \frac{b - a}{2^n} \tag{5}$$

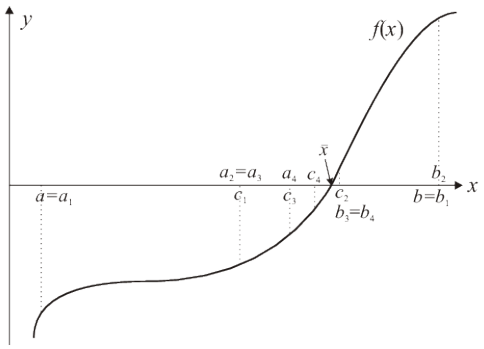
   So the method converges since in the limit $n \to \infty$ the factor $2^{-n} \to 0$, but as we see it converges rather slowly.

5. If our desired maximum error ϵ, say $\epsilon = 10^{-4}$, is reached, we stop

$$|c_n - \bar{x}| \leq \epsilon \;\Rightarrow\; \frac{b - a}{2^n} \leq \epsilon \tag{6}$$

   Along with the above convergence criteria, we can also test if $|f(c)| < \epsilon$ since at root x_0 implies $f(x_0) = 0$.

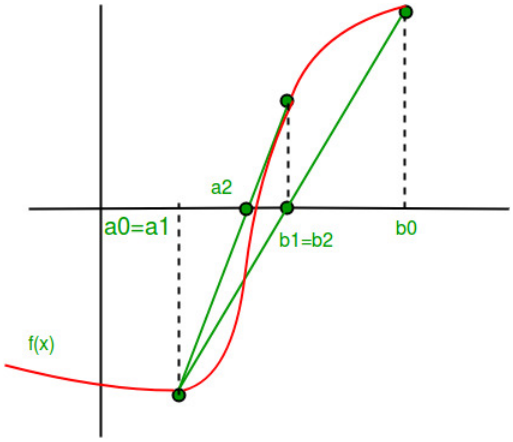
The bisection steps are schematically shown in the figure below.

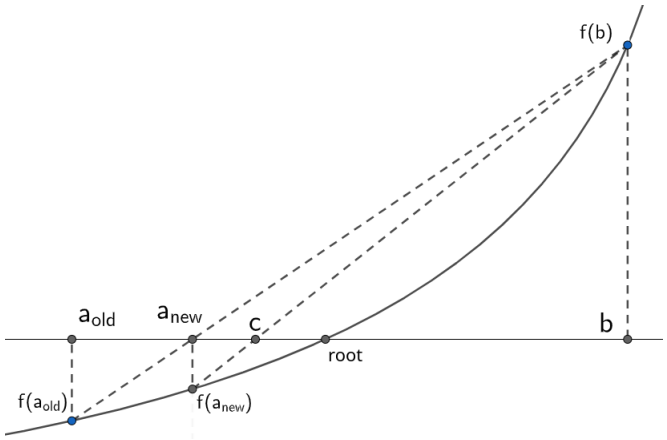


This method has slowest convergence of all other root finding methods but it is a sure shot to root provided you can bracket properly. But still one cannot get root beyond certain precision because the difference between b and a is limited by floating point precision *i.e.* as the difference $(b_n - a_n)$ decreases. Therefore, the accuracy can never reach machine precision.

Regula falsi method

In Regula Falsi or False Position method, one does some sort of an interpolation to converge on a root faster than Bisection. The method involves finding the slope of the straight line joining $[a_1, b_1]$, which bracketed the root and finding where this line crosses the abscissa (x-axis) and take that point as either new a_2 or b_2 depending on the relative sign of the functions at those points. One important difference over Bisection method is that the new starting point is directly determined by the function $f(x)$ under consideration apart from checking $f(a) * f(b) \lesseqgtr 0$. Two possible scenario is schematically shown in the figure below.





The basic steps involved are

1. Choose appropriate $[a, b]$, where $a < b$ and $f(a) * f(b) < 0$, to bracket the root. Details about bracketing the root have already been discussed before in Bisection method.

2. Calculate the slope of the straight line joining a and b and obtain c where the line crosses the abscissa *i.e.* $y(c) = 0$,

$$m = \frac{y(b) - y(a)}{b - a} = \frac{y(b) - y(c)}{b - c} \quad \Rightarrow \quad c = b - \frac{(b - a) * y(b)}{y(b) - y(a)} \tag{7}$$

   where $y(a) = f(a)$ and $y(b) = f(b)$. One can as well use $f(a)$ as reference point instead of $f(b)$, as is shown in the figure above. If the function $f(x)$ is convex or concave, as in the right figure, in the interval $[a, b]$ containing a root, then one of the points a or b is always fixed and the other point varies with iterations. This sometime makes checking for convergence little tricky. Hence, after n-th step,

$$c_n = b_n - \frac{(b_n - a_n) * f(b_n)}{f(b_n) - f(a_n)} \tag{8}$$

3. If $f(a_n) * f(c_n) < 0$, then root lies to the left of c_n and in such case the new $b_{n+1} = c_n$ and $a_{n+1} = a_n$. If $|c_{n-1} - c_n| < \epsilon$ then c_n is the root implying $f(c_n) \approx 0$. Else iterate step 2 with new b.

4. If $f(a_n) * f(c_n) > 0$, then root lies to the right of c_n, therefore, the new $a_{n+1} = c_n$ and $b_{n+1} = b_n$. If $|c_{n-1} - c_n| < \epsilon$ then c_n is the root implying $f(c_n) \approx 0$. Else iterate step 2 with new a.

The Regula falsi always converges and has improved speed of convergence over Bisection, but the rate of convergence can sometimes drop below Bisection. Please note that

as a solution is approached, a and b will be very close to each other and subtraction in the denominator of (7) can lose significant digits. This method is slightly different than the *Secant method* where it retains the last two computed points that are obtained the same formula as above.

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 $f'(x)$ but does not require bracketing. 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 an initial guess x_0 for a root of $f(x)$,

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{1}{2!}(x - x_0)^2 f''(x_0) + \cdots \tag{9}$$

If we are closer to the root, then $(x - x_0)^2 \approx 0$ and we can stop at $f'(x)$ term,

$$f(x) = f(x_0) + (x - x_0)f'(x_0) = 0 \quad \Rightarrow \quad x = x_0 - \frac{f(x_0)}{f'(x_0)} \tag{10}$$

then x is a better approximation of the root than x_0. Far from a root, the higher derivative terms in the series become important. The approximation to the root can be improved iteratively to move from the x_0 towards the root,

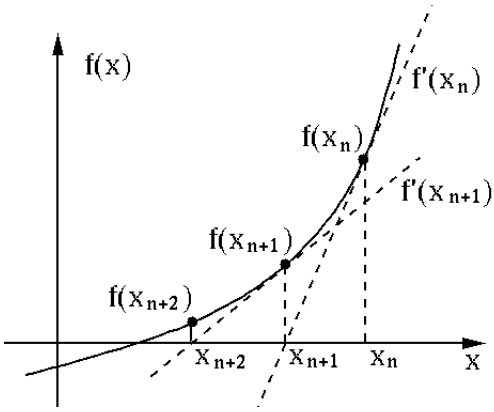
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \qquad n = 1, 2, \ldots \tag{11}$$

There is no bracketing here, just an initial guess x_0 but involved taking a derivative. This can be a problem unless we have an analytical expression for the derivative and cheaper to evaluate. When Newton-Raphson works it converges quadratically but that often is not the case. For a detailed discussion on convergence of Newton-Raphson see Wikipedia or textbooks including Numerical Recipes.

In place of using analytical expression for derivative $f'(x)$ of the function $f(x)$, one can approximate the derivative with finite difference, resulting in a variant of Newton's method called Secant method.

$$f'(x) \approx \frac{f(x + h) - f(x - h)}{2h} \tag{12}$$

The symmetric derivative is preferred over forward $([f(x + h) - f(x)]/h)$ or backward $([f(x)-f(x-h)]h)$ derivatives simply because it is $\mathcal{O}(h^3)$ imported. Therefore the use of the finite difference formula in (12) requires two initial guesses x_0 and x_1 corresponding to $x \pm h$.



The steps of Newton-Raphson method is fairly straight forward,

1. Make a good guess of x_0
2. Evaluate $f(x)$ and its derivative $f'(x)$ at $x = x_0$.
3. Continue using (11) to improve the estimate of the root until $|x_{n+1} - x_n| < \epsilon$.

Laguerre's method: roots of polynomials

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

$$P(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n = (x - x_1)\,(x - x_2) \cdots (x - x_n) \qquad (13)$$

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 $P(x)$,

$$P(x) = (x - x_1)\, Q(x) \qquad (14)$$

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 (14) to the next lower degree polynomial and so one till the end.

$$\begin{aligned} P(x) &= (x - x_1)\, Q(x) \\ Q(x) &= (x - x_2)\, R(x) \quad \Rightarrow \quad P(x) = (x - x_1)(x - x_2)\, R(x) \ \text{ etc.} \end{aligned} \qquad (15)$$

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 given in (14).

3. Else calculate the following

$$G = \frac{P'(\alpha_n)}{P(\alpha_n)}, \quad H = G^2 - \frac{P''(\alpha_n)}{P(\alpha_n)} \quad \Rightarrow \quad a = \frac{n}{G \pm \sqrt{(n-1)(nH - G^2)}} \tag{16}$$

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

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

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

6. Go for deflation (15) 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.

	-1	3	0	-4
2	$+$	-2	2	4
	-1	1	2	0

   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 $(x - x_n)$.

	-1	1	2
2	$+$	-2	-2
	-1	-1	0

$$\Rightarrow \quad Q(x) = -x - 1$$

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