Roots of nonlinear equations

Consider solving a nonlinear equation in one variable x

$$f(x) = 0$$

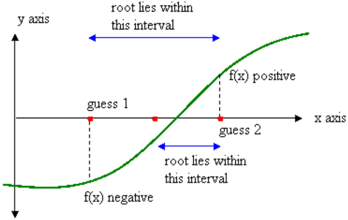
Assume $f(x)$ smoothly varying, continuous over the range $[a, b]$.

A few typical examples of $f(x)$ are,

$$f(x) = \cos x - x^3 \;/\; 3x + \sin x - \exp x \;/\; x \exp x - 2 \;/\; x^3 + 3x - 5 \text{ etc.}$$

If x_0 in the interval $[a_0, b_0]$ satisfies equation $f(x_0) = 0$, then x_0 is a *root* or *zero* of the function and is **one** of the solutions in that interval.

Since $f(x)$ is continuous and $[a_0, b_0]$ so chosen such that $f(a_0)$ and $f(b_0)$ are of opposite signs, then according to *intermediate value theorem*, $f(x)$ has at least one root in the interval $[a_0, b_0]$.



Finding root numerically starts with guess $[a_0, b_0]$ – informed or trial-and-error – at which $f(x)$ has opposite signs.

$[a_0, b_0]$ are said to *bracket* the root.

Iterations proceed by producing a sequence of shrinking intervals $[a_0, b_0] \rightarrow [a_i, b_i]$ – shrunk intervals always contain one root of $f(x)$.

For convergence, necessary to have a good initial guess – *(i)* plotting $f(x)$ vs x or *(ii)* informed expectation.

Algorithms for finding roots of nonlinear equations :

1. Bisection method
2. False position (Regula falsi) method
3. Newton-Raphson method
4. Laguerre's method (for roots of polynomials)

As is evident, the first three methods are iterative and, therefore, call for user specific precision ϵ, typically 10^{-4}.

Bisection method

Simplest, relatively slower but guaranteed to converge provided the bracketing is done carefully – $f(a)$ and $f(b)$ have opposite signs at the interval boundary $[a, b]$ and $f(x)$ is continuous.

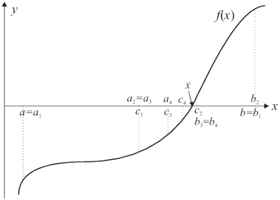
Things can go wrong – *(i)* when $f(x) = 0$ is an extrema or *(ii)* multiple roots in the interval chosen.

Steps involve in achieving bracketing :

1. Choose a_0 and b_0, and calculate $f(a_0)$ and $f(b_0)$.
2. If $f(a_0) \star f(b_0) < 0$, bracketing done, proceed to execute *bisection*.
3. If $f(a_0) * f(b_0) > 0$ *i.e.* same sign, check whether $|f(a_0)| \lessgtr |f(b_0)|$.
   3.1 If $|f(a_0)| < |f(b_0)|$, shift a_0 further left by $a_0' = a_0 - \beta(b_0 - a_0)$ and go back to step 2. Choose your own β, say 1.5.
   3.2 If $|f(a_0)| > |f(b_0)|$, shift b_0 further right by $b_0' = b_0 + \beta(b_0 - a_0)$ and go back to step 2. Choose your own β, say 1.5.
4. If the condition $f(a_0) \star f(b_0) < 0$ is not met in $10 - 12$ iterations, start with a new pair $[\bar{a}_0, \bar{b}_0]$ and back to step 2.

Steps involve in bisection method :

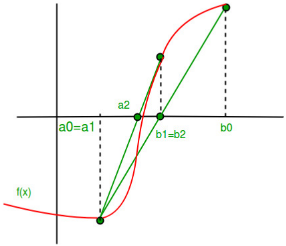
1. Choose bracket $[a_0, b_0]$, where $a_0 < b_0$ and $f(a_0) * f(b_0) < 0$. If $|b_0 - a_0| < \epsilon$, done! Check also $f(a_0)$ and/or $f(b_0) < \delta$.
2. If none of the conditions are met, bisect the interval $c_0 = (a_0 + b_0)/2$. Choose new interval $[a_0, c_0] \equiv [a_1, b_1]$ if $f(c_0) \star f(a_0) < 0$ OR set $[c_0, b_0] \equiv [a_1, b_1]$ if $f(c_0) \star f(b_0) < 0$. Check for $|b_1 - a_1| < \epsilon$ along with $f(a_1$ or $b_1) < \delta$.
3. If not, keep on bisecting the interval and re-adjusting the interval till $|b_n - a_n| < \epsilon$ along with $f(a_n$ or $b_n) < \delta$ are satisfied.
4. Bisecting n times, we have a possible solution in the interval length
$$\frac{|b_n - a_n|}{2} = \frac{|b_0 - a_0|}{2^n} \leq \epsilon$$
Accuracy can never reach machine precision – limited by floating point precision with decreasing $(b_n - a_n)$

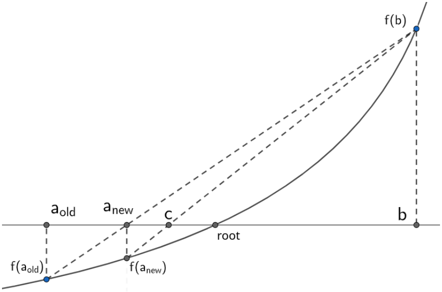


Regula falsi method

Interpolation to converge on a root faster than Bisection –

Find slope of the straight line joining $[a_0, b_0]$ that has bracketed the root. The point c_0 where this straight line crosses $x - axis$ is the new a or b depending on sign of $f(c_0)$.





Unlike Bisection, new interval boundary a_n or b_n directly given by $f(x)$.

Always converges and has improved speed of convergence. Here too, as (a, b) get close, can lose significant digits.

Steps involve in Regula falsi :

1. Choose bracket $[a_0, b_0]$, where $a_0 < b_0$ and $f(a) * f(b) < 0$. If $|b_0 - a_0| < \epsilon$, done! Check also $f(a_0)$ and/or $f(b_0) < \delta$.
2. Calculate slope of the line joining a_0 and b_0 and obtain c_0 where the line crosses abscissa $y(c_0) = 0$,

$$m = \frac{f(b_0) - f(a_0)}{b_0 - a_0} = \frac{f(b_0) - f(c_0)}{b_0 - c_0} \Rightarrow c_0 = b_0 - \frac{(b_0 - a_0) * f(b_0)}{f(b_0) - f(a_0)}$$

   Reference point can as well be $f(a_0)$. If $f(x)$ is convex or concave, then one of the points a, b is fixed and the other varies with iterations. After n-th step,

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

3. If $f(a_n) * f(c_n) < 0$, then root lies to left of $c_n \Rightarrow b_{n+1} = c_n$ and $a_{n+1} = a_n$. If $|c_{n-1} - c_n| < \epsilon$ then c_n is the root $f(c_n) \approx 0$.
4. If $f(b_n) * f(c_n) < 0$, then root lies to right of $c_n \Rightarrow a_{n+1} = c_n$ and $b_{n+1} = b_n$. If $|c_{n-1} - c_n| < \epsilon$ then c_n is the root $f(c_n) \approx 0$.

Newton-Raphson method

Involves both $f(x)$ and $f'(x)$ but does not require bracketing. Method is based on Taylor's expansion.

Works for multivariate functions. Converges quadratically $\Rightarrow$ near root the number of significant digits approximately doubles with each step.

To solve $f(x) = 0$, Taylor expand $f(x)$ around initial guess x_0,

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

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

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

then x is better approximation of root than x_0 but involves taking derivative. Useful if derivative is cheaper to evaluate and hence to code.

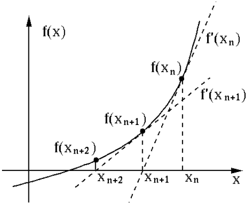
Approximation to the root can be improved iteratively to move from the x_0 towards the root,

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

Alternatively, use finite difference to approximate derivative – a variant of Newton's method called Secant method,

$$f'(x) \approx \frac{1}{2h}\left[f(x+h) - f(x-h)\right] + \mathcal{O}(h^3)$$

Finite difference requires two initial guesses x_0, x_1 corresponding to $x \pm h$.



The steps of Newton-Raphson method are

1. Make a good guess of x_0.
2. Evaluate $f(x)$ and $f'(x)$ at $x = x_0$.
3. Use iterative updating $x_n \to x_{n+1}$

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

4. Continue to improve the estimate of the root until $|x_{n+1} - x_n| < \epsilon$ and/or $f(x_n) \approx \delta$.

Solve the following nonlinear equations in the interval $[-1.5, 1.5]$:

$$x - 2\cos(x) = 0, \qquad \cos(x) - x^3 = 0$$
$$3x + \sin(x) - e^x = 0, \qquad x\,e^x - 2 = 0$$

Laguerre's method

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

$$P(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n = (x - \alpha_1)(x - \alpha_2)\cdots(x - \alpha_n)$$

Polynomials with real a_i, roots can be real or complex conjugate pair. We will restrict ourselves to only real roots.

A two step method –
(i) Laguerre's method followed by *(ii)* deflating the polynomial.

Using Laguerre's method determine the root α_1 of $P(x)$, then obtain a reduced polynomial $Q(x)$ of degree one less than $P(x)$.

$$\begin{aligned} P(x) &= (x - \alpha_1)\, Q(x) \\ Q(x) &= (x - \alpha_2)\, R(x) \quad \Rightarrow \quad P(x) = (x - \alpha_1)(x - \alpha_2)\, R(x) \ \text{ etc.} \end{aligned}$$

where the roots of $Q(x)$ are the remaining roots of $P(x)$. In each of n steps, the Laguerre determines the roots α_i while deflation determines the remainder polynomial.

Method used for deflation is synthetic division method (learnt in class IX).

Laguerre algorithm proceeds as

1. Begin with an initial guess β_0.
2. If β_0 is bang on one of the roots, *i.e.* $P(\beta_0) \approx 0$, go for deflation.
3. Else calculate the following

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

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

4. Set $\beta_{k+1} = \beta_k - a$ as new trial.
5. Iterate till $|\beta_{k+1} - \beta_k| < \epsilon$ and set $\alpha_1 = \beta_k$. Check $P(\alpha_1) \approx 0$.
6. Go for deflation to reduce the degree of the polynomial and do the above iteration all over again to find α_2 and so on.

Next is deflation, divide $P(x)$ by $(x - \alpha_1)$, then $Q(x)$ by $(x - x_2)$ and so on.

Deflation by Synthetic division method :

1. Arrange terms in $P(x)$ in ascending order of power. Store the coefficients, the leading one be 1 and missing coefficient(s) are 0.

$$\frac{P(x)}{x-\alpha_1} = \frac{-x^3+3x^2-4}{x-2} \Rightarrow \text{divisor} = 2, \ \text{coeffs} = [-1,\ 3,\ 0,\ -4]$$

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. Multiply it again with the divisor and 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 α_1 is a root then the last sum, which gives the remainder, must be zero. The reduced lower degree polynomial $Q(x)$ has the numbers below the line as coefficients

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

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

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

The polynomial in the example is thus factorized in terms of its roots as

$$P(x) = -x^3 + 3x^2 - 4 = (x-2)(x-2)(-x-1)$$

Find all the roots of the following polynomial

$P(x) = 6x^3 - 11x^2 - 26x + 15$ answer : $x = 3, -5/3, 1/2$

$P(x) = x^4 - x^3 - 7x^2 + x + 6$ answer : $x = 1, -2, 3, -1$