5. Numerical methods for solving nonlinear equations in $\mathbb{R}$

5.1. One-step methods

Let $f : \Omega \to \mathbb{R}$, $\Omega \subset \mathbb{R}$. Consider the equation

$$f(x) = 0, \quad x \in \Omega. \tag{1}$$

Particular cases.

1) Case $m = 2$.

$$F_2^T(x_i) = x_i - \frac{f(x_i)}{f'(x_i)}.$$

This method is called **Newton's method** (**the tangent method**). Its order is 2.

2) Case $m = 3$.

$$F_3^T(x_i) = x_i - \frac{f(x_i)}{f'(x_i)} - \frac{1}{2}\left[\frac{f(x_i)}{f'(x_i)}\right]^2 \frac{f''(x_i)}{f'(x_i)},$$

with $\text{ord}(F_3^T) = 3$. So, this method converges faster than F_2^T.

3) Case $m = 4$.

$$F_4^T(x_i) = x_i - \frac{f(x_i)}{f'(x_i)} - \frac{1}{2}\frac{f''(x_i)f^2(x_i)}{[f'(x_i)]^3} + \frac{\left(f'''(x_i)f'(x_i) - 3[f''(x_i)]^2\right)f^3(x_i)}{3![f'(x_i)]^5}.$$

Remark 1 *The higher the order of a method is, the faster the method converges. Still, this doesn't mean that a higher order method is more efficient (computation requirements). By the contrary, the most efficient are the methods of relatively low order, due to their low complexity (methods F_2^T and F_3^T).*

According to [Remark 14, Cs.9], there always exists a neighborhood of α where the $F-$method is convergent. Choosing x_0 in such a neighborhood allows approximating α by terms of the sequence

$$x_{i+1} = F_2^T(x_i) = x_i - \frac{f(x_i)}{f'(x_i)}, \quad i = 0, 1, ...,$$

with a prescribed error ε.

If α is a solution of equation (1) and $x_{n+1} = F_2^T(x_n)$, for approximation error, [Remark 17, Cs. 9] gives

$$\left|\alpha - x_{n+1}\right| \leq \frac{1}{2}[f(x_n)]^2 M_2 g.$$

Lemma 2 *Let $\alpha \in (a, b)$ be a solution of equation (1) and let $x_n = F_2^T(x_{n-1})$. Then*

$$|\alpha - x_n| \leq \frac{1}{m_1}|f(x_n)|, \quad \textit{with } m_1 \leq m_1 f = \min_{a \leq x \leq b}\left|f'(x)\right|.$$

Proof. We use the mean formula

$$f(\alpha) - f(x_n) = f'(\xi)(\alpha - x_n),$$

with $\xi \in$ to the interval determined by α and x_n. From $f(\alpha) = 0$ and $|f'(x)| \geq m_1$ for $x \in (a, b)$, it follows $|f(x_n)| \geq m_1 |\alpha - x_n|$, that is

$$|\alpha - x_n| \leq \frac{1}{m_1}|f(x_n)|.$$

■

In practical applications the following evaluation is more useful:

Lemma 3 *If $f \in C^2[a, b]$ and F_2^T is convergent, then there exists $n_0 \in \mathbb{N}$ such that*

$$|x_n - \alpha| \leq |x_n - x_{n-1}|, \quad n > n_0.$$

Remark 4 *The starting value is chosen randomly. If, after a fixed number of iterations the required precision is not achieved, i.e., condition $|x_n - x_{n-1}| \leq \varepsilon$, does not hold for a prescribed positive ε, the computation has to be started over with a new starting value.*

A modified form of Newton's method: - the same value during the computation of f':

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_0)}, \quad k = 0, 1, \ldots.$$

It is very useful because it doesn't request the computation of f' at x_j, $j = 1, 2, \ldots$ but the order is no longer equal to 2.

Another way for obtaining Newton's method.

We start with x_0 as an initial guess, sufficiently close to the α. Next approximation x_1 is the point at which the tangent line to f at $(x_0, f(x_0))$ crosses the Ox-axis. The value x_1 is much closer to the root α than x_0.

We write the equation of the tangent line at $(x_0, f(x_0))$:

$$y - f(x_0) = f'(x_0)(x - x_0).$$

If $x = x_1$ is the point where this line intersects the Ox-axis, then $y = 0$

$$-f(x_0) = f'(x_0)(x_1 - x_0),$$

and solving for x_1 gives

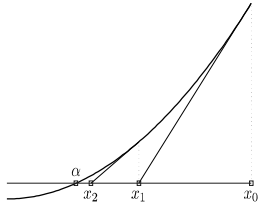
$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

By repeating the process using the tangent line at $(x_1, f(x_1))$, we obtain for x_2

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

For the general case we have

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



The algorithm:

Let x_0 be the initial approximation.

for $n = 0, 1, ..., ITMAX$

$$x_{n+1} \leftarrow x_n - \frac{f(x_n)}{f'(x_n)}.$$

A stopping criterion is:

$$|f(x_n)| \leq \varepsilon \text{ or } \left|x_{n+1} - x_n\right| \leq \varepsilon \text{ or } \frac{\left|x_{n+1} - x_n\right|}{\left|x_{n+1}\right|} \leq \varepsilon,$$

where ε is a specified tolerance value.

Example 5 *Use Newton's method to compute a root of* $x^3 - x^2 - 1 = 0$, *to an accuracy of* 10^{-4}. *Use* $x_0 = 1$.

Sol. *The derivative of f is* $f'(x) = 3x^2 - 2x$. *Using* $x_0 = 1$ *gives* $f(1) = -1$ *and* $f'(1) = 1$ *and so the first Newton's iterate is*

$$x_1 = 1 - \frac{-1}{1} = 2 \text{ and } f(2) = 3, \ f'(2) = 8.$$

The next iterate is

$$x_2 = 2 - \frac{3}{8} = 1.625.$$

Continuing in this manner we obtain the sequence of approximations which converges to 1.465571.

5.2. Multistep methods for solving nonlinear eq. in $\mathbb{R}$

Let $f:\Omega\to\mathbb{R}$, $\Omega\subset\mathbb{R}$. Consider the equation

$$f(x)=0,\quad x\in\Omega, \tag{3}$$

We attach a mapping $F:D\to\Omega,\ D\subset\Omega^n$ to this equation.

Let $(x_0,...,x_n)\in D$ be *the starting points*. We construct iteratively the sequence

$$x_0,x_1,...,x_{n-1},x_n,x_{n+1}... \tag{4}$$

with

$$x_i=F(x_{i-n-1},...,x_{i-1}),\quad i=n+1,.... \tag{5}$$

The problem consists in choosing F and $x_0,...,x_n\in D$ such that the sequence (4) to be convergent to the solution of the equation (3).

The F-method is **a multistep method**.

It is based on interpolation methods with more than one interpolation node.

Let $\alpha\in\Omega$ be a solution of equation (3), let $(a,b)\subset\Omega$ be a neighborhood of α that isolates this solution and $x_0,...,x_n\in(a,b)$, some given values.

Denote by g the inverse function of f, assuming it exists. Because $\alpha=g(0)$, the problem reduces to approximating g by an interpolation method with $n>1$ nodes, for example Lagrange, Hermite, Birkhoff, etc...

Lagrange inverse interpolation

Let $y_k=f(x_k)$, $k=0,...,n$, hence $x_k=g(y_k)$. We attach the Lagrange interpolation formula to y_k and $g(y_k)$, $k=0,...,n$:

$$g=L_ng+R_ng, \tag{6}$$

where

$$(L_ng)(y)=\sum_{k=0}^{n}\frac{(y-y_0)...(y-y_{k-1})(y-y_{k+1})...(y-y_n)}{(y_k-y_0)...(y_k-y_{k-1})(y_k-y_{k+1})...(y_k-y_n)}g(y_k). \tag{7}$$

Taking

$$F_n^L(x_0,...,x_n)=(L_ng)(0),$$

F_n^L is a $(n+1)-$ steps method defined by

$$\begin{aligned}F_n^L(x_0,...,x_n)&=\sum_{k=0}^{n}\frac{y_0...y_{k-1}y_{k+1}...y_n}{(y_k-y_0)...(y_k-y_{k-1})(y_k-y_{k+1})...(y_k-y_n)}(-1)^ng(y_k)\\&=\sum_{k=0}^{n}\frac{y_0...y_{k-1}y_{k+1}...y_n}{(y_k-y_0)...(y_k-y_{k-1})(y_k-y_{k+1})...(y_k-y_n)}(-1)^nx_k.\end{aligned}$$

Concerning the convergence of this method we state:

Theorem 6 *If $\alpha\in(a,b)$ is solution of equation (3), f' is bounded on (a,b), and the starting values satisfy*

$$|\alpha-x_k|<1/c,\quad k=0,...,n,$$

with $c=$constant, then the sequence

$$x_{i+1}=F_n^L(x_{n-i},...,x_i),\quad i=n,n+1,...$$

converges to α.

Remark 7 *The order $ord(F_n^L)$ is the positive solution of the equation*

$$t^{n+1}-t^n-...-t-1=0.$$

Particular cases.

1) For $n=1$, the nodes x_0,x_1, we get **the secant method**

$$F_1^L(x_0,x_1)=x_1-\frac{(x_1-x_0)f(x_1)}{f(x_1)-f(x_0)},$$

Thus,

$$x_{k+1} := F_1^L(x_{k-1}, x_k) = x_k - \frac{(x_k - x_{k-1})\, f(x_k)}{f(x_k) - f(x_{k-1})}, \quad k = 1, 2, ...$$

is the new approximation obtained using the previous approximations x_{k-1}, x_k.

The *order* of this method is the positive solution of equation:

$$t^2 - t - 1 = 0,$$

so $ord(F_1^L) = \frac{(1+\sqrt{5})}{2}$.

A modified form of the secant method: if we keep x_1 fixed and we change every time the same interpolation node, i.e.,

$$x_{k+1} = x_k - \frac{(x_k - x_1)\, f(x_k)}{f(x_k) - f(x_1)}, \quad k = 2, 3, \ldots.$$

2) For $n = 2$, the nodes x_0, x_1, x_2 and we get

$$F_2^L(x_0, x_1, x_2) = \frac{x_0 f(x_1) f(x_2)}{[f(x_0) - f(x_1)][f(x_0) - f(x_2)]} + \frac{x_1 f(x_0) f(x_2)}{[f(x_1) - f(x_0)][f(x_1) - f(x_2)]} + \frac{x_2 f(x_0) f(x_1)}{[f(x_2) - f(x_0)][f(x_2) - f(x_1)]}.$$

The *order* of this method is the positive solution of equation:

$$t^3 - t^2 - t - 1 = 0,$$

so $ord(F_2^L) = 1.8394$.

Another way of obtaining secant method.

Based on approx. the function by a straight line connecting two points on the graph of f (not required f to have opposite signs at the initial points).

The first point, x_2, of the iteration is taken to be the point of intersection of the Ox-axis and the secant line connecting two starting points $(x_0, f(x_0))$ and $(x_1, f(x_1))$. The next point, x_3, is generated by the intersection of the new secant line, joining $(x_1, f(x_1))$ and $(x_2, f(x_2))$ with the Ox-axis. The new point, x_3, together with x_2, is used to generate the next point, x_4, and so on.

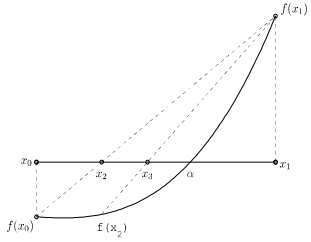
The formula for x_{n+1} is obtained by setting $x = x_{n+1}$ and $y = 0$ in the equation of the secant line from $(x_{n-1}, f(x_{n-1}))$ to $(x_n, f(x_n))$:

$$\frac{x - x_n}{x_{n-1} - x_n} = \frac{y - f(x_n)}{f(x_{n-1}) - f(x_n)} \Leftrightarrow x = x_n + \frac{(x_{n-1} - x_n)(y - f(x_n))}{f(x_{n-1}) - f(x_n)},$$

we get

$$x_{n+1} = x_n - f(x_n)\left[\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}\right]. \tag{8}$$

Note that x_{n+1} depends on the two previous elements of the sequence $\Rightarrow$ two initial guesses, x_0 and x_1,for generating $x_2, x_3, \ldots$.



The algorithm:

Let x_0 and x_1 be two initial approximations.

for $n = 1, 2, ..., ITMAX$

$$x_{n+1} \leftarrow x_n - f(x_n)\left[\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}\right].$$

A suitable stopping criterion is

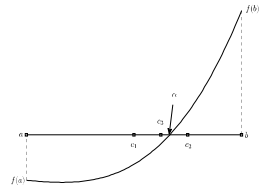
$$|f(x_n)| \le \varepsilon \text{ or } \left|x_{n+1} - x_n\right| \le \varepsilon \text{ or } \frac{\left|x_{n+1} - x_n\right|}{\left|x_{n+1}\right|} \le \varepsilon,$$

where ε is a specified tolerance value.

Example 8 *Use the secant method with* $x_0 = 1$ *and* $x_1 = 2$ *to solve* $x^3 - x^2 - 1 = 0$, *with* $\varepsilon = 10^{-4}$.

Sol. *With* $x_0 = 1$, $f(x_0) = -1$ *and* $x_1 = 2$, $f(x_1) = 3$, *we have*

$$x_2 = 2 - \frac{(2-1)(3)}{3 - (-1)} = 1.25$$



Bisection method

from which $f(x_2) = f(1.25) = -0.609375$. *The next iterate is*

$$x_3 = 1.25 - \frac{(1.25-2)(-0.609375)}{-0.609375-3} = 1.3766234.$$

Continuing in this manner the iterations lead to the approximation 1.4655713.

Examples of other multi-step methods

1. THE BISECTION METHOD

Let f be a given function, continuous on an interval $[a, b]$, such that

$$f(a)f(b) < 0. \tag{9}$$

By Mean Value Theorem, it follows that there exists at least one zero α of f in (a, b).

The bisection method is based on halving the interval $[a, b]$ to determine a smaller and smaller interval within α must lie.

First we give the midpoint of $[a, b]$, $c = (a + b)/2$ and then compute the product $f(c)f(b)$. If the product is negative, then the root is in the interval $[c, b]$ and we take $a_1 = c$, $b_1 = b$. If the product is positive, then the root is in the interval $[a, c]$ and we take $a_1 = a$, $b_1 = c$. Thus, a new interval containing α is obtained.

The algorithm:

Suppose $f(a)f(b) \leq 0$. Let $a_o = a$ and $b_o = b$.

for $n = 0, 1, ...,$ITMAX

$c \leftarrow \frac{a_n + b_n}{2}$

if $f(a_n)f(c) \leq 0$, set $a_{n+1} = a_n, b_{n+1} = c$

else, set $a_{n+1} = c, b_{n+1} = b_n$

The process of halving the new interval continues until the root is located as accurately as desired, namely

$$|a_n - b_n| < \varepsilon, \tag{10}$$

where a_n and b_n are the endpoints of the n-th interval $[a_n, b_n]$ and ε is a specified precision. The approximation of the solution will be $\frac{a_n + b_n}{2}$.

Some other stopping criterions: $\frac{|a_n - b_n|}{|a_n|} < \varepsilon$ or $|f(a_n)| < \varepsilon$.

Example 9 *The function* $f(x) = x^3 - x^2 - 1$ *has one zero in* $[1, 2]$. *Use the bisection algorithm to approximate the zero of* f *with precision* 10^{-4}.

Sol. *Since* $f(1) = -1 < 0$ *and* $f(2) = 3 > 0$, *then (9) is satisfied. Starting with* $a_0 = 1$ *and* $b_0 = 2$, *we compute*

$$c_0 = \frac{a_0 + b_0}{2} = \frac{1+2}{2} = 1.5 \text{ and } f(c_0) = 0.125.$$

Since $f(1.5)f(2) > 0$, *the function changes sign on* $[a_0, c_0] = [1, 1.5]$.

To continue, we set $a_1 = a_0$ *and* $b_1 = c_0$; *so*

$$c_1 = \frac{a_1 + b_1}{2} = \frac{1 + 1.5}{2} = 1.25 \text{ and } f(c_1) = -0.609375$$

Again, $f(1.25)f(1.5) < 0$ *so the function changes sign on* $[c_1, b_1] = [1.25,\ 1.5]$. *Next we set* $a_2 = c_1$ *and* $b_2 = b_1$. *Continuing in this manner we obtain a sequence* $(c_i)_{i>0}$ *which converges to* 1.465454, *the solution of the equation.*

2. THE METHOD OF FALSE POSITION

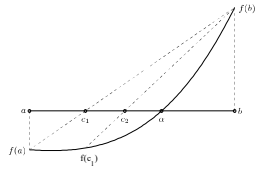
This method is also known as *regula falsi*, is similar to the Bisection method but has the advantage of being slightly faster than the latter. The function have to be continuous on $[a, b]$ with

$$f(a)f(b) < 0.$$

The point c is selected as point of intersection of the Ox-axis, and the straight line joining the points $(a,\ f(a))$ and $(b, f(b))$. From the equation of the secant line, it follows that

$$c = b - f(b)\frac{b-a}{f(b)-f(a)} = \frac{af(b) - bf(a)}{f(b) - f(a)} \tag{11}$$

Compute $f(c)$ and repeat the procedure between the values at which the function changes sign, that is, if $f(a)f(c) < 0$ set $b = c$, otherwise set $a = c$. At each step we get a new interval that contains a root of f and the generated sequence of points will eventually converge to the root.



Method of false position.

The algorithm:

Given a function f continuous on $[a_0, b_0]$, with $f(a_0)f(b_0) < 0$,

input: a_0, b_0

for $n = 0, 1, ..., ITMAX$

$c \leftarrow \frac{f(b_n)a_n - f(a_n)b_n}{f(b_n) - f(a_n)}$

if $f(a_n)f(c) < 0$, set $a_{n+1} = a_n, b_{n+1} = c$ **else** set $a_{n+1} = c, b_{n+1} = b_n$.

Stopping criterions: $|f(a_n)| \leq \varepsilon$ or $|a_n - a_{n-1}| \leq \varepsilon$, where ε is a specified tolerance value.

One of the main disadvantages of this method is that if the sequence of points generated by its algorithm is one-sided, the convergence of the method is slow.

Remark 10 *The bisection and the false position methods converge at a very low speed compared to the secant method.*

Example 11 *The function* $f(x) = x^3 - x^2 - 1$ *has one zero in* $[1, 2]$. *Use the method of false position to approximate the zero of* f *to within* 10^{-4}.

Sol. *A root lies in the interval* $[1, 2]$ *since* $f(1) = -1$ *and* $f(2) = 3$. *Starting with* $a_0 = 1$ *and* $b_0 = 2$, *we get using (11)*

$$c_0 = 2 - \frac{3(2-1)}{3-(-1)} = 1.25 \text{ and } f(c_0) = -0.609375.$$

Here, $f(c_0)$ *has the same sign as* $f(a_0)$ *and so the root must lie on the interval* $[c_0, b_0] = [1.25, 2]$. *Next we set* $a_1 = c_0$ *and* $b_1 = b_0$ *to get the next approximation*

$$c_1 = 2 - \frac{3 - (2 - 1.25)}{3 - (-0.609375)} = 1.37662337 \text{ and } f(c_1) = -0.2862640.$$

Now $f(x)$ *change sign on* $[c_1, b_1] = [1.37662337, 2]$. *Thus we set* $a_2 = c_1$ *and* $b_2 = b_1$. *Continuing in this manner the iterations lead to the approximation* 1.465558.