CHAPTER II
Finding Roots of Real Single Valued Functions

Nabil R. Nassif and Dolly K. Fayad

February, 2013

In this chapter we consider one of the most encountered problems in scientific computing, which is the problem of computing the **root or zero** of a real-valued function f. This is in short equivalent to computing the solution of a nonlinear equation starting from one or several **initial data** , adopting some **iterative method** that - under favorable conditions - will **converge** to a zero of the function f.

1 Introduction

Let f be a real-valued function of a real variable admitting a specific regularity on its domain D, i.e., let f be k-times continuously differentiable, with $k \geq 1$ ($f \in C^k(D)$). We seek to find the roots of this function f, defined as follows:

Definition 1 *The set R of roots of the function $f(x)$ is defined as:*

$$R = \{r \in \mathbb{R} : \ f(r) = 0\}.$$

Given some computational tolerance $\epsilon_{tol} = \frac{1}{2}10^{1-m}$, $m = 1, 2, ...$, our objective is to compute one or more roots of f, within such ϵ_{tol}. Specifically, for any $r \in R$, we seek an approximation r_a to r, ($r_a \approx r$), such that:

$$\frac{|r - r_a|}{|r|} \leq \epsilon_{tol}. \tag{1}$$

(We say then, that r_a approximates r up to m decimal places)

The search for a specific root of a function requires two steps.

1. **Step 1: Locate the root,** i.e. seek an interval (a, b), with $O(|b - a|) = O(|r|)$, such that:

$$f(x) \in C([a, b]), (\text{i.e. } f(x) \text{ is at least continuous}) \tag{2}$$
$$r \in (a, b) \tag{3}$$
$$f(a) \times f(b) < 0 \tag{4}$$
$$\forall\, x \in (a, b),\ x \neq r \ \Rightarrow\ f(x) \neq 0 \tag{5}$$

2. **Step 2:** Generate a sequential process leading to a **sequence** $\{r_n\}_{n\geq 0}$ the terms of which are in (a, b) and that converges to r, i.e. satisfying:

$$r_n \in (a, b)\, \forall\, n \text{ and } \lim_{n\to\infty} r_n = r. \tag{6}$$

   The generation of such a sequence is usually done through an **iterative** procedure (or method) where $r_n = g(r_{n-1}, ..., r_{n-k})$, $k \geq 1$.

We start by introducing some general properties verified by such methods.

Definition 2 *A numerical method is said to be a* **one-step** *method in case $k = 1$, the initial state of the sequence being determined by the only choice of r_0; otherwise, it is a* **multi-step** *method of order k, and its initial state is then determined by the choice of $r_0, .., r_{k-1}$.*

The **order of convergence** of a method measures the rate at which the sequence $\{r_n\}$ generated by the numerical process converges to the root r. It is defined as follows:

Definition 3 Order of Convergence of a Method
A method is of order $\alpha > 0$, if there exists a sequence of positive numbers $\{t_n\}_{n\geq 0}$, such that $\forall n \geq 1$:

$$|r - r_n| \leq t_n, \text{ with } t_n \leq C t_{n-1}^{\alpha} \tag{7}$$

Equivalently, in the special case where $t_n = |r - r_n|$:

$$|r - r_n| \leq C|r - r_{n-1}|^{\alpha} \tag{8}$$

The constants C and α are independent from n, with $C < 1$ for $n = 1$.

If $\alpha = 1$ the convergence is said to be **linear**, while if $\alpha > 1$ the convergence is **super-linear**. In particular, if $\alpha = 2$ the convergence of the method is **quadratic**. (Note also that the greatest α is, the fastest is the method.)

Definition 4 Global convergence versus Local convergence
A method is said to be **globally** *convergent if the generated sequence $\{r_n\}_n$ converges to r for any choice of the initial state; otherwise it is* **locally** *convergent.*

When implemented, the process generating the elements of $\{r_n\}$ will be stopped as soon as the 1^{st} computed element r_{n_0} satisfies some predefined "stopping criteria".

Definition 5 Stopping criteria
Given some tolerance ϵ_{tol}, a standard stopping criterion is defined by the following relative estimates:

$$\frac{|r_{n_0} - r_{n_0-1}|}{|r_{n_0}|} \leq \epsilon_{tol} \text{ and } \frac{|r_n - r_{n-1}|}{|r_n|} > \epsilon_{tol} \text{ if } n < n_0. \tag{9}$$

The "remainder" $f(r_n)$ can also be used to set a stopping criterion since $f(r) = \lim_{n\to\infty} f(r_n) = 0$. Thus, one may use a relative evaluation of the remainder. Specifically, find the first element r_{n_0} of the sequence $\{r_n\}$ satisfying:

$$\frac{|f(r_{n_0})|}{|f(r_0)|} \leq \epsilon_{tol} \text{ and } \frac{|f(r_n)|}{|f(r_0)|} > \epsilon_{tol} \text{ if } n < n_0. \tag{10}$$

Note also that by using the Mean-Value Theorem one has:

$$0 = f(r) = f(r_n) + (r - r_n)f'(c_n), \text{ where } c_n = r + \theta(r_n - r),\ \theta \in (0, 1).$$

Thus if f' is available (referring also to (9)), a more sophisticated stopping criterion would be:

$$\frac{|f(r_{n_0})|}{|r_{n_0} f'(r_{n_0})|} \leq \epsilon_{tol} \text{ and } \frac{|f(r_n)|}{|r_{n_0} f'(r_{n_0})|} > \epsilon_{tol} \text{ if } n < n_0. \tag{11}$$

In this chapter, we shall analyze successively three root finding iterative methods: the Bisection method, Newton's method and the Secant method

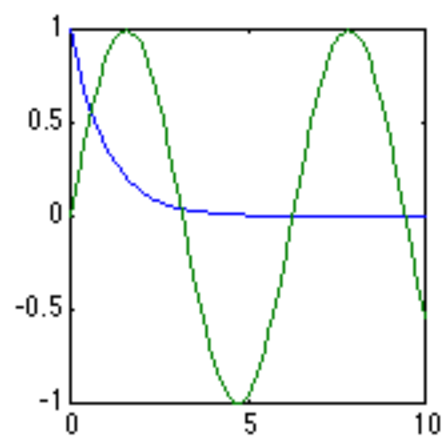
2 How to locate the roots of a function

There are basically two approaches to **locate the roots** of a function f. The first one seeks to **analyze the behaviour of** f analytically or through plotting its graph, while the second one transforms the problem of root finding into an equivalent **fixed point problem**. We illustrate these methods through some specific examples.

Example 1 *Locate the roots of the function* $f(x) = e^{-x} - \sin(x)$.

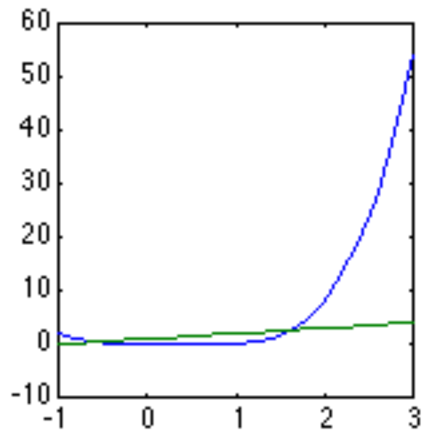
1. **Analyzing the behaviour of the function**
   Since for all $x < 0$ the exponential $e^{-x} > 1$, then one concludes that $f(x) > 0$. Furthermore $f(0) = 1$. This implies that all the roots of the function lie in the interval $(0, \infty)$. Moreover, studying the variation of the function $f(x)$ is done by studying the sign of its derivative.

2. **A fixed point problem**
   Let $g_1(x) = e^{-x}$ and $g_2(x) = \sin x$. Solving the problem $f(r) = 0$ can be made equivalent to solving the equation $g_1(r) = g_2(r)$, r becoming a fixed-point for these 2 functions . Hence plotting them on the same graph, one concludes straightforwardly that g_1 and g_2 intersect at an infinite number of points with positive abcissa, that constitute the set of all roots of f.



Example 2 *Locate the roots of the quadratic polynomial* $p(x) = x^4 - x^3 - x - 1$

To use the fixed point method, let $g_1(x) = x^4 - x^3$ and $g_2(x) = x + 1$. It is easy to verify in this case that these 2 functions intersect twice, implying consequently that f has 2 roots located respectively in the intervals $(-1, 0)$ and $(1, 2)$.



3 The Bisection Method

The **Bisection method** is a procedure that repeatedly **"halves"** the interval in which a root r has been located. This "halving" process is reiterated until the desired accuracy is reached. Specifically, after locating the root in (a, b) we proceed as follows:

- The **midpoint** of (a, b), $r_1 = \frac{a+b}{2}$ and $y = f(r_1)$ are computed. If it happens fortuitously that $f(r_1) = 0$ then the root has been found, i.e. $r = r_1$. Otherwise $y \neq 0$ and 2 cases may occur:
  - either $y \times f(a) < 0$, implying that $r \in (a, r_1)$
  - or $y \times f(a) > 0$, in which case $r \in (r_1, b)$.

  Let the initial interval $(a, b) = (a_0, b_0)$.
  Either way, and as a consequence of this first halving of (a_0, b_0), one obtains a new interval $(a_1, b_1) = (a_0, r_1)$ or $(a_1, b_1) = (r_1, b_0)$, such that one obviously has:

  (12) $$r \in (a_1, b_1), \text{ with } b_1 - a_1 = \frac{1}{2}(b_0 - a_0) \text{ and } |r - r_1| \leq (b_1 - a_1).$$

- Evidently this process can be repeated, generating a sequence of intervals $\{(a_n, b_n) | n \geq 1\}$ such that:

  (13) $$r \in (a_n, b_n) \text{ with } b_n - a_n = \frac{1}{2}(b_{n-1} - a_{n-1})$$

and a sequence of iterates $\{r_n \,|\, n \geq 1\}$, with $r_n \in (a, b)\, \forall n$, and where

$$r_n = \frac{1}{2}(a_{n-1} + b_{n-1}) \text{ with } |r - r_n| \leq (b_n - a_n). \tag{14}$$

- The process is achieved when the interval (a_n, b_n) is relatively small with respect to the initial interval, specifically when the least value of n is reached, for which:

$$\frac{b_n - a_n}{b_0 - a_0} \leq \epsilon_{tol} \tag{15}$$

  where ϵ_{tol} is a given computational tolerance.

At the end of this process, the best estimate of the root r would be the last computed value of r_n as in (14).
The Bisection Method is implemented through the following algorithm:

Algorithm 1 Bisection Method

```
function[r,n]=myBisection(f,a,b,tol,kmax)
% Inputs:  f, a, b, kmax, tol
% kmax: maximum acceptable  number of iterations; tol=0.5*10^(-p+1)
%  S: stopping criteria = [length last (a,b)] / [length initial (a, b)]
% Outputs: r : sequence of midpoints  converging to the root , and
%  n: number of iterations used within tol.
fa=f(a);
% length of initial interval (a,b)
ab=abs(b-a);
% Initialize n and S
n=1;S=1;
while S>tol & n<kmax
  r(n)=(a+b)/2;y=f(r(n));
    if y*fa<0
       b=r(n);
    elseif  y*fa>0
       a=r(n);fa=y;
    elseif y*fa=0
       disp(' r(n) is the root' )
       break
    end
S=(abs(b-a)/ab);
n=n+1;
end
%If n>=kmax, reconsider the values allocated to the parameters: a, b, S, kmax
if n>=kmax
disp ( ' error  no convergence' ) ;
end
```

The parameter $kmax$ is used as a programming safeguard. This eliminates the possibility of entering an infinite loop in case the sequence diverges, or also when the program is incorrectly coded. If k exceeds $kmax$ with $\frac{b_k-a_k}{b-a} > tol$, the written algorithm would then signal an error.

Thus, (12), (13) and (14) lead to the following result.

Theorem 1 *Under assumptions (2)-(5), the bisection algorithm generates 2 sequences $\{a_n\}_{n\geq 0}$ and $\{b_n\}_{n\geq 0}$ from which one "extracts" a sequence of iterates $\{r_n\}_{n\geq 1}$, with $r_n = a_n$ or $r_n = b_n$, such that:*

1. $a_0 = a$, $b_0 = b$,
2. $r \in (a_n, b_n)$ *with* $a_n < r < b_n$, $\forall n \geq 0$,
3. *The sequences $\{a_n\}$ and $\{b_n\}$ are respectively monotone increasing and decreasing,*
4. $b_n - a_n = \frac{b_{n-1}-a_{n-1}}{2} = \frac{b-a}{2^n}\,\forall\, n \geq 1$, *and* $\lim_{n\to\infty} a_n = \lim_{n\to\infty} b_n = r$,
5. $|r - r_n| \leq b_n - a_n$, $\forall\, n \geq 1$.

Proof. 1. and 2. are obtained by construction.
To prove 3., given (a_{n-1}, b_{n-1}) with $r \in (a_{n-1}, b_{n-1})$, then by definition of the method, $r_n = \frac{1}{2}(a_{n-1} + b_{n-1})$ will either be a_n or b_n. Therefore, in the case the process is reiterated, this implies that either $a_n = a_{n-1}$ and $b_n < b_{n-1}$ or $a_n > a_{n-1}$ and $b_n = b_{n-1}$ which proves the required result. (Note that neither of these sequences can "stagnate". For example, the existence of an n_0 such that $a_{n_0} = a_n$, $\forall n \geq n_0$, would imply that $r = a_{n_0}$, i.e. the process is finite and the root has been found after n_0 steps !).
4. follows from the "halving" procedure. It can be easily shown by induction, that $b_n - a_n = \frac{b-a}{2^n}$ and therefore $\lim_{n\to\infty} b_n - a_n = 0$, meaning that the sequences of lengths $\{(b_n - a_n)\}$ of the intervals $\{(a_n, b_n)\}$ converge to 0. Hence, the sequences $\{a_n\}$ and $\{b_n\}$ have the same limit point r.
Finally, to obtain 5., just note again that $r_n = a_n$ or b_n, with $r \in (a_n, b_n)$. ■

A consequence of these properties is the linearity of the convergence and an estimate on the minimum number of iterations needed to achieve a given computational tolerance ϵ_{tol}. Specifically, we have the following result.

Corollary 1 *Under the assumptions of the previous theorem, one obtains the following properties:*

1. *The bisection method converges linearly, in the sense of definition (3), i.e.*

$$|r - r_n| \leq t_n = b_n - a_n, \; with\, t_n \leq \frac{1}{2} t_{n-1}$$

2. *The minimum number of iterations needed to reach a tolerance of* $\epsilon_{tol} = 0.5 \times 10^{1-p}$ *is given by*

$$k = \lceil (p-1)\frac{\ln(10)}{\ln(2)} + 1 \rceil$$

Proof. The first part of the corollary is a direct result from the previous theorem. As for the second part, it is achieved by noting that the method reaches the desired accuracy, according to the selected stopping criteria, whenever n reaches the value k such that:

$$\frac{b_k - a_k}{b - a} \leq \epsilon_{tol} < \frac{b_{k-1} - a_{k-1}}{b - a} < ... \frac{b_1 - a_1}{b - a} = \frac{1}{2} < \frac{b_0 - a_0}{b - a} = 1. \tag{16}$$

From equation (16) and since $\frac{b_n - a_n}{b-a} = \frac{1}{2^n}\ \forall\ n \geq 0$, we can estimate the **least number of iterations** required (theoretically) to reach the relative precision $\epsilon_{tol} = \frac{1}{2}10^{1-p}$, p being the number of significant decimal figures fixed by the user. Such integer k satisfies then:

$$\frac{1}{2^k} \leq \frac{1}{2}10^{1-p} < \frac{1}{2^{k-1}}. \tag{17}$$

Equivalently:

$$-k\ln(2) \leq (1-p)\ln(10) - \ln(2) < -(k-1)\ln(2),$$

from which one concludes that:

$$k\ln(2) \geq (p-1)\ln(10) + \ln(2) > (k-1)\ln(2),$$

leading to:

$$k \geq (p-1)\frac{\ln(10)}{\ln(2)} + 1 > k-1, \tag{18}$$

The integer k is computed then as:

$$k = \lceil (p-1)\frac{\ln(10)}{\ln(2)} + 1 \rceil.$$

■

Note that such k is independent of a and b, since it estimates the ratio $\frac{b_k - a_k}{b-a}$, a measure of the relative reduction of the size of the interval (a_k, b_k) containing r.
The following table provides values of k relative to a fixed precision p.

precision p	**Iterations** k
3	8
5	15
7	21
10	31
15	48

Obviously the method is slowly convergent ! Nevertheless, since at each step the length of the interval is reduced by a factor of 2, it is advantageous to choose the initial interval as small as possible.
In applying the bisection method algorithm for the above 2 examples, one gets the following results:

1. Let $f(x) = e^{-x} - \sin(x)$. Search for the root of f in the interval $[0, 1]$, with a tolerance $\epsilon = 0.5 \times 10^{-5}$ (6 significant figures rounded).

Iteration	**Iterate**
1	$5.000000\ 10^{-1}$
...	....
10	$5.888672\ 10^{-1}$
11	$5.885009\ 10^{-1}$
12	$5.886230\ 10^{-1}$
13	$5.885010\ 10^{-1}$
14	$5.885620\ 10^{-1}$
15	$5.885315\ 10^{-1}$
16	$5.885468\ 10^{-1}$
17	$5.885391\ 10^{-1}$
18	$5.885353\ 10^{-1}$
19	$5.885334\ 10^{-1}$
20	$5.885324\ 10^{-1}$
21	$5.885329\ 10^{-1}$

The Bisection method took 20 iterations to reach a precision of 6. The $21st$ was needed to meet the termination condition.

2. Let $f(x) = x^4 - x^3 - x - 1$. Search for the root of f in the interval $[0, 3]$ with $\epsilon = 0.5 \times 10^{-4}$ (5 significant figures rounded).

Iteration	**Iterate**
1	$1.500000\ 10^{0}$
2	$2.250000\ 10^{0}$
...	....
10	$1.620118\ 10^{0}$
11	$1.618653\ 10^{0}$
12	$1.617921\ 10^{0}$
13	$1.618287\ 10^{0}$
14	$1.618104\ 10^{0}$
15	$1.618013\ 10^{0}$
16	$1.618059\ 10^{0}$

To conclude, the Bisection is a **multi-step method** that, although conceptually clear and simple, has significant drawbacks since it is relatively slow. However it **globally converges** to the searched solution and is often used as a starter for other more efficient **locally convergent** methods, notably both Newton's and Secant methods.

n	a_n	b_n	r_n	$f(a_n) \times f(b_n)$
0	0	1	0.5	+
1	0.5	1	0.75	+
2	0.75	1	0.875	-
3	0.75	0.875	0.813	-
4	0.75	0.813	0.782	-
5	0.75	0.782	0.766	-
6	0.75	0.766	0.758	+
7	0.758	0.766	0.762	+
8	0.762	0.766	0.764	-
9	0.762	0.764	0.763	+
10	0.763	0.764	0.763	

4 Newton's Method

Newton's (or the **Newton-Raphson's**) method is one of the most powerful numerical methods for solving non-linear equations. It is also referred to as the **tangent method**, as it consists in constructing a sequence of numbers $\{r_n | r_n \in (a, b) \,\forall n \geq 1\}$, obtained by intersecting tangents to the curve $y = f(x)$ at the sequence of points $\{(r_{n-1}, f(r_{n-1}))| \, n \geq 1\}$ with the $x-$ axis. Constructing such tangents and such sequences requires additional assumptions to (2)-(5) as derived hereafter.
To start, let $r_0 \in (a, b)$ in which the root is located, and let $M_0 = (r_0, f(r_0))$ be the point on the curve

$$\{(\mathcal{C})|y = f(x),\, a \leq x \leq b\}.$$

Let also $(\mathcal{T}_0)$ be the tangent to $(\mathcal{C})$ at M_0 with equation given by:

$$y = f'(r_0)(x - r_0) + f(r_0).$$

The intersection of $(\mathcal{T}_0)$ with the $x-$axis is obtained for $y = 0$ and is given by:

$$r_1 = r_0 - \frac{f(r_0)}{f'(r_0)}. \tag{19}$$

To insure that $r_1 \in (a, b)$, r_0 should be chosen "close enough" to r. Specifically, since $f(r) = 0$, (19) is equivalent to:

$$r_1 - r = r_0 - r - \frac{f(r_0) - f(r)}{f'(r_0)} \tag{20}$$

Using Taylor's expansion of $f(r)$ about r_0 up to first order , one has:

$$f(r) = f(r_0) + f^{'}(r_0)(r - r_0) + \frac{1}{2} f^{''}(c_0)(r - r_0)^2,\; c_0 = r_0 + \theta_0(r - r_0),\; 0 < \theta_0 < 1,$$

thus leading to:

$$\frac{f(r)-f(r_0)}{f'(r_0)} = (r-r_0) + \frac{1}{2}\frac{f''(c_0)}{f'(r_0)}(r-r_0)^2, \text{ with } c_0 \in (a,b).$$

Hence, imposing on f and on the interval (a,b) the following additional assumptions:

(21) $$f(x) \in C^2(a,b), \text{i.e. } f(x),\ f'(x),\ f''(x) \text{ are continuous on } (a,b)$$

(22) $$f'(x) \neq 0\ \forall x \in (a,b)$$

ones concludes from (20):

(23) $$|r_1 - r| = \frac{1}{2}\frac{|f''(c_0)|}{|f'(r_0)|}(r-r_0)^2$$

Based on these additional assumptions, we define also the positive constant:

(24) $$C = \frac{1}{2}\frac{\max_{x\in(a,b)}|f''(x)|}{\min_{x\in(a,b)}|f'(x)|}.$$

which will then lead to:

(25) $$|r - r_1| \leq C|r-r_0|^2$$

This gives a preliminary "closeness" result of r_1 with respect to the root r, in terms of the "closeness" of r_0, without however insuring yet the required location of r_1 in (a,b). In this view, letting now:

(26) $$I_0 = \{x|\,|r-x| < \frac{1}{C}\} \cap (a,b)$$

and selecting initially r_0 in I_0, leads to the required result as shown hereafter.

Lemma 1 *If $r_0 \in I_0$ as defined in (26), then $r_1 \in I_0$ with*

(27) $$|r - r_1| \leq |r - r_0|$$

Proof. Let $e_i = C|r - r_i|$, $i = 0,1$, where C verifies (24). Multiplying (25) by C one obviously has:

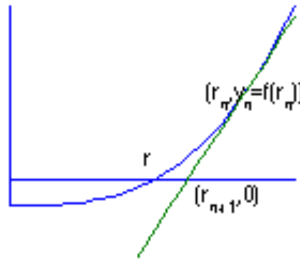
$$e_1 \leq e_0^2$$

moreover, since $e_0 < 1$ and $C > 0$, the required result is reached. ■

Thus selecting $r_0 \in I_0$ and reaching r_1 satisfying (27), the process can be continued beyond that step. In fact one generates a sequence of **Newton's** iterates $\{r_n|\, n \geq 2\}$ with $r_n \in (a,b)\,\forall n$, given by a formula generalizing (19). Specifically, one has:

(28) $$r_{n+1} = r_n - \frac{f(r_n)}{f'(r_n)},\ n \geq 0.$$

with $(r_{n+1}, 0)$ being the intersection with the x-axis of the tangent to the curve $(\mathcal{C})$ at the point $(r_n, f(r_n))$.



Clearly, Newton's method is a one-step iteration $r_{n+1} = g(r_n)$, with the **iteration** function $g(x)$ given by:

$$g(x) = x - \frac{f(x)}{f'(x)}. \tag{29}$$

We turn now to the analysis of the convergence of Newton's method, i.e. the convergence of Newton's iterates $\{r_n\}_{n\geq 0}$.

Theorem 2 *Let $f(x)$ satisfies assumptions (2)-(5), in addition to (21) and (22), then for $r_0 \in I_0$, with C as defined in (24), the sequence of Newton's iterates:*

$$r_{n+1} = r_n - \frac{f(r_n)}{f'(r_n)},\ \ n \geq 0,$$

is such that:

1. $r_n \in I_0$, $\forall n \geq 0$
2. $\lim_{n\to\infty} r_n = r$
3. $|r - r_{n+1}| \leq C|r - r_n|^2$, *meaning that Newton's method is quadratic with $\alpha = 2$.*

Proof. The proof of this theorem follows from arguments used to obtain lemma 1. In fact, one derives as for (25) that:

$$e_{n+1} \leq e_n^2,\ \forall n \geq 0. \tag{30}$$

where $e_i = C|r - r_i|,\ i = n, n+1$.
Moreover, it can be easily proved by induction on n, that (30) in turn implies that:

$$e_n \leq (e_0)^{2^n}\ \forall\ n \geq 1. \tag{31}$$

As $e_0 < 1$ then $r_n \in I_0$ with $\lim_{n\to\infty} e_n = 0$, proving parts 1 and 2 of the lemma. In addition to these results, and as derived in (23) and (25), one concludes that :

$$|r_{n+1} - r| = \frac{1}{2}\frac{|f''(c_n)|}{|f'(r_n)|}(r - r_n)^2 \leq C|r - r_n|^2, \tag{32}$$

with $c_n = r_n + \theta_n(r - r_n)$, $0 < \theta_n < 1$. Referring to (8) that result obviously implies that $\alpha = 2$, ■

Note also that inequality (31) allows obtaining an estimate on the minimum number of iterations needed to reach a computational tolerance $\epsilon_{tol} = 0.5 \times 10^{1-p}$. Specifically, we prove now:

Corollary 2 *If $r_0 \in I_0$, the minimum number of iterations needed to reach $\epsilon_{tol} = 0.5 \times 10^{1-p}$ is given by:*

$$n_0 = \lceil \ln(1 + \frac{(p-1)\ln(10) + \ln(2)}{|\ln(e_0)|})/\ln(2)\rceil.$$

Proof. Note that ϵ_{tol} is reached whenever $n = n_0$ satisfies the following inequalities:

$$\frac{|r - r_{n_0}|}{|r - r_0|} \leq 0.5 \times 10^{1-p} < \frac{|r - r_n|}{|r - r_0|}, \forall n < n_0,$$

Since also $\frac{|r-r_n|}{|r-r_0|} = \frac{e_n}{e_0}$, $\forall n \geq 1$, then from (31):

$$\frac{|r - r_{n_0}|}{|r - r_0|} \leq (e_0)^{2^{n_0}-1}.$$

The sought for minimum number of iterations n_0 would thus verify:

$$(e_0)^{2^{n_0}-1} \leq 0.5 \times 10^{1-p} < (e_0)^{2^n-1}, \forall n < n_0.$$

Since $e_0 < 1$, this is equivalent to:

$$2^{n_0} \geq 1 + \frac{(p-1)\ln(10) + \ln(2)}{|\ln(e_0)|} > 2^n, \; n < n_0.$$

This leads to n_0 satifying:

$$n_0 \geq \frac{\ln(1 + \frac{(p-1)\ln(10)+\ln(2)}{|\ln(e_0)|})}{\ln(2)} > n_0 - 1$$

and therefore:

$$n_0 = \lceil \ln(1 + \frac{(p-1)\ln(10) + \ln(2)}{|\ln(e_0)|})/\ln(2)\rceil,$$

which is the desired result. ■

To illustrate, assume $e_0 = \frac{1}{2}$, then it results from this lemma that:

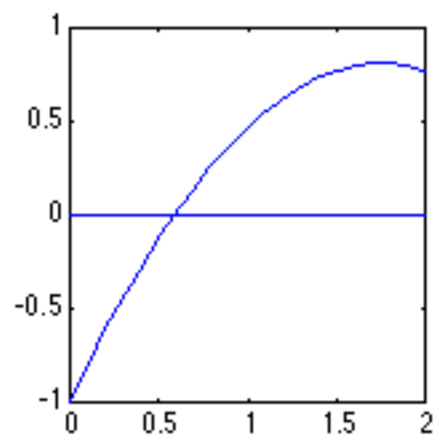
$$n_0 = \lceil \ln(2 + (p-1)\frac{\ln(10)}{\ln(2)})/\ln(2)\rceil.$$

The following table provides values of n_0 relative to a precision p.

precision p	**Iterations** n_0
7	4
10	5
16	6

Thus, one can assert that Newton's method is a **locally** and **quadratically convergent** method. When a root r of a function $f(x)$ is located in an interval (a, b), the first step is to insure finding a sub-interval $I_0 \subset (a, b)$ containing r, in which $|r - r_0| \leq \frac{1}{C}$, with the constant C given in (24).
A rule of thumb would be to select r_0 after 1 or 2 applications of the bisection method. Such a step would make sure the initial condition r_0 is close "enough" to r.
The following example illustrates the general behavior of Newton's method.

Example 3 *Find the roots of* $f(x) = \sin(x) - e^{-x}$ *in the interval* $(0, 2)$, *using Newton's method.*



Obviously, Newton's method is not applicable when the initial choice of the iteration r_0 is selected from the whole interval $(0, 2)$. For example if r_0 is chosen in the interval $(1.5, 2)$, the generated sequence $\{r_n\}$ may not fall in the interval $(0, 2)$ and thus fails to converge, as is shown in the following table resulting from the application of Newton's algorithm with $r_0 = 1.75$.

Iteration	**Iterate**
0	1.75
1	1.8291987×10^2
2	1.8206468×10^2
3	1.8221346×10^2
4	1.8221237×10^2
...	...

Obviously, the convergence is taking place to a root that **is not** in the interval $(0, 2)$. On the other hand, one application of the bisection method would start the iteration with $r_0 = 1$, leading to the following efficiently convergent process.

Iteration	Iterate
0	1.0
1	$4.785277889803116\times10^{-1}$
2	$5.841570194114709\times10^{-1}$
3	$5.885251122073911\times10^{-1}$
4	$5.885327439585476\times10^{-1}$
5	$5.885327439818611\times10^{-1}$
6	$5.885327439818611\times10^{-1}$

Obviously, about 4 iterations would provide 10 significant figures, a fifth one leading to 16 figures, i.e. a more than a double precision answer.

However, there are cases, as in the first Example below, where the convergence of the method is not affected by the choice of the initial condition, whereby Newton's method converges unconditionally.

Example 4 The square root function

Using Newton's method, we seek an approximation to $r = \sqrt{a}$, where $a > 0$..
Clearly, such r is the unique positive root of $f(x) = x^2 - a$, with Newton's iterates satisfying the following identity:

$$r_{n+1} = r_n - \frac{f(r_n)}{f'(r_n)} \equiv \frac{1}{2}(r_n + \frac{a}{r_n}),\ \forall\, n \geq 0 \tag{33}$$

(It is easy to check graphically that the sequence converge to $\sqrt{a}$ for any initial choice of $r_0 > 0$).
Based on the equation above:

$$r_{n+1} - r = \frac{1}{2}(r_n - 2r + \frac{a}{r_n})$$

Equivalently, since $a = r^2$:

$$r_{n+1} - r = \frac{(r_n - r)^2}{2r_n} \geq 0 \tag{34}$$

The following results can therefore be deduced:

1. $r_n \geq r,\ \forall\ n \geq 1$
2. The generated iterative sequence $\{r_n\}$ is a decreasing sequence, since:

   $$r_{n+1} - r_n = -\frac{f(r_n)}{f'(r_n)} = -\frac{(r_n^2 - r^2)}{2r_n} \leq 0$$

   based on the property 1. above.

3. The sequence $\{r_n\}$ converges to the root of f, i.e. $\lim_{n\to\infty} r_n = r$, since rewriting (34) as:
$$r_{n+1} - r = \frac{r_n - r}{2}(1 - \frac{r}{r_n})$$
in turn by induction leads to:
$$r_{n+1} - r < \frac{1}{2}(r_n - r) < ... < \frac{1}{2^{(n-1)}}(r_1 - r)$$

4. The convergence is notably quadratic, since from (34) and for all $n \geq 0$:
$$|r_{n+1} - r| = |\frac{(r_n - r)^2}{2r_n}| < C|r_n - r|^2 \text{ where } C = \frac{1}{2r}$$

As for IEEE standard notations, note that since
$$a = m \times 2^e \ \text{ with } 1 \leq m < 2$$
then the square root function is such that:
$$\sqrt{\ } : (m, e) \to (m', e') \text{ with } \sqrt{a} = m' \times 2^{e'}$$
The normalized mantissa and exponent of $\sqrt{a}$ are computed as follows:

1. if $e = 2k$, then $m' = \sqrt{m}$ with $1 \leq m' < \sqrt{2} < 2$, and $e' = k$, i.e.
$$\sqrt{\ } : (m, e = 2k) \to (m' = \sqrt{m}, e' = k)$$

2. if $e = 2k + 1$, then $a = 2m \times 2^{2k}$ and $m' = \sqrt{2m}$ with $1 < \sqrt{2} \leq m' < 2$, and $e' = k$, i.e.
$$\sqrt{\ } : (m, e = 2k + 1) \to (m' = \sqrt{2m}, e' = k)$$

In either case, Newton's iteration in binary mode may start with $r_0 = 1$. ■

The local character of convergence of Newton's method is well illustrated in the interesting case of the reciprocal function.

Example 5 The reciprocal of a positive number function
Assume $a > 0$. We seek an approximation to $r = \frac{1}{a}$, where r is the unique positive root of $f(x) = a - \frac{1}{x}$. Obviously, Newton's iterations satisfy the following identity:

(35) $$r_{n+1} = r_n(2 - ar_n), \ \forall\ n \geq 0$$

Choosing restrictively the initial condition $r_0 \in (0, 2/a)$ leads to an iterative sequence $\{r_n\}$ where:

(36) $$r_{n+1} > 0, \text{ whenever } r_n \in (0, 2/a)$$

In such case, for all $n \geq 1$, it is left as an exercise to prove that:

1. $r_{n+1} - r = -\frac{(r_n - r)^2}{r}$
2. The generated sequence is an increasing sequence
3. The sequence $\{r_n\}$ converges to the root of f, i.e. $\lim_{n\to\infty} r_n = r$
4. Convergence of the sequence is quadratic.

Considering IEEE standard notations as for the square root function example, if

$$a = m \times 2^e \text{ , with } 1 < m < 2$$

then the inverse function is such that:

$$inv \ : (m, e) \to (m', e'), \text{ with } \frac{1}{a} = m' \times 2^{e'}$$

The normalized mantissa and exponent of $1/a$ are respectively:

$$m' = 2/m \text{ and } e' = -e - 1$$

since $\frac{1}{a} = \frac{1}{m} \times 2^{-e}$ or more adequately:

$$\frac{1}{a} = \frac{2}{m} \times 2^{-e-1}, \text{ with } 1 < \frac{2}{m} < 2$$

■

5 The Secant Method

Recall that Newton's iteration satisfies formula (28):

$$r_{n+1} = r_n - \frac{f(r_n)}{f'(r_n)} \text{ where } f'(r_n) = \lim_{h\to 0} \frac{f(r_n + h) - f(r_n)}{h}$$

One drawback of Newton's method is the necessary availability of the derivative. In case such function is difficult to program, an alternative would be to avoid the calculation of $f'(r_n)$, and replace it by the backward divided difference approximation to the derivative:

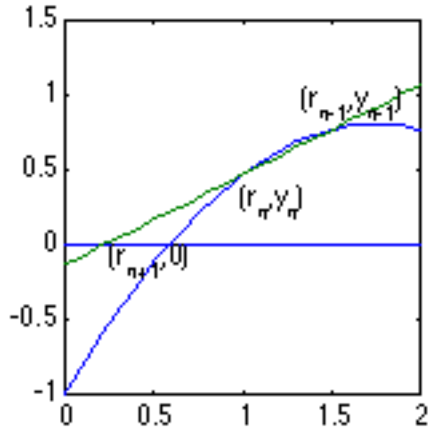
$$f'(r_n) \approx [r_{n-1}, r_n] = \frac{f(r_n) - f(r_{n-1})}{r_n - r_{n-1}}$$

This would suggest obtaining r_{n+1} using the secant to the curve $y = f(x)$ passing through the points $(r_{n-1}, f(r_{n-1}))$ and $(r_n, f(r_n))$, the equation of which is:

$$y = \frac{f(r_n) - f(r_{n-1})}{r_n - r_{n-1}}(x - r_n) + f(r_n)$$

The intersection of this secant line with the $x-$ axis would provide the $(n+1)$-iterate **secant method** formula:

$$r_{n+1} = r_n - \frac{f(r_n)}{[r_n, r_{n-1}]} \equiv r_n - \frac{f(r_n)(r_n - r_{n-1})}{f(r_n) - f(r_{n-1})}, n \geq 2 \qquad (37)$$



The secant method is a two-steps method of the form $r_{n+1} = g(r_n, r_{n-1})$, its processing requiring selection of r_0 and r_1. Of course, if the method is succeeding, the points r_n will be approaching a zero of f, so $f(r_n)$ will be converging to zero.
Practically, if a root r of the function f is located in the interval (a, b), one would suggest applying twice the bisection method in order to implement (37) as shown in the following algorithm.

Algorithm 2 Secant Method

```
% Input f, a, b,TOL, kMAX
% Find the first 2 approximations by the ‘‘Bisection rule"
function[r,k]=mySecant(f,a,b,TOL,kmax)
r(1)=(a+b)/2 ;
if f(a)*f(r(1)) < 0
    r(2)=(r(1)+a)/2 ;
else
   r(2)=(b+r(1))/2 ;
end
k=2; S = 1;
while S >TOL & k<=kMAX
     d=(f(r(k))-f(r(k-1))/r(k)-r(k-1));
     r(k+1)=r(k)-f(r(k))/d;
     S = abs (r(k+1)-r(k)])  abs (r(k)) ;
     k=k+1;
end
```

The advantages of the secant method relative to the tangent method are that (after the first step) only one function evaluation is required per step (in contrast to Newton's iteration which requires 2) and that it is almost as rapidly convergent. It can be shown that under the same assumptions as those of Theorem 2 , the basic secant method is **superlinear** and has a **local character** of convergence.

Theorem 3 *Under the hypothesis of Theorem 2 and for r_0 and $r_1 \in I_0$ (defined in (26)), then one has:*

1. $\lim_{n\to\infty} r_n = r$,
2. *There exists a sequence $\{t_n|,\ n \geq 0\}$ such that:*

(38) $$|r - r_n| \leq t_n, \ \text{with } t_n = O(t_{n-1})^\gamma \ \text{ and } \gamma = \frac{1+\sqrt{5}}{2}$$

i.e the **order of convergence** *of the secant method is the* **Golden Number** $\gamma \approx$ **1.618034** *in the sense of (7).*

Proof. Starting with the following identity (Theorem 4.5) :

$$f(r) = f(r_n) + [r_{n-1}, r_n](r - r_n) + \frac{1}{2}(r - r_n)(r - r_{n-1})f''(c) \ ; \ c = r_n + \theta(r - r_n),\ 0 < \theta < 1$$

where $f(r) = 0$, one deduces:

$$r_n - \frac{f(r_n)}{[r_{n-1}, r_n]} = r + \frac{1}{2}(r - r_n)(r - r_{n-1})\frac{f''(c)}{[r_{n-1}, r_n]}$$

Since $[r_{n-1}, r_n] = f'(c_1)$, then under the assumptions of Theorem 2 , one concludes that:

(39) $$|r - r_n| \leq C|r - r_{n-1}|.|r - r_{n-2}|,\ \forall n \geq 2$$

with C as defined in (24). Again, let $e_i = C|r - r_i|,\ i = n-2, n-1$, then (39) is equivalent to:

(40) $$e_n \leq e_{n-1}.e_{n-2},\ \forall n \geq 2.$$

With the assumption that the initial conditions r_0, r_1 are selected so that:

(41) $$\delta = \max(e_0, e_1) < 1$$

one obviously concludes that $e_2 \leq e_0 e_1 < \delta^2$ and that $e_3 \leq e_1 e_2 < \delta^3$. Let $\{f_n|n \geq 0\}$ be a Fibonacci sequence defined by:

$$f_0 = f_1 = 1,\ f_n = f_{n-1} + f_{n-2},\ n \geq 2.$$

It is well known that the solution of this second order difference equation is given by:

$$f_n = \frac{1}{\sqrt{5}}((\frac{1+\sqrt{5}}{2})^{n+1} - (\frac{1-\sqrt{5}}{2})^{n+1}) = \frac{1}{\sqrt{5}}((\frac{1+\sqrt{5}}{2})^{n+1} + (-1)^{n+1}(\frac{\sqrt{5}-1}{2})^{n+1}).$$

Let $\gamma = \frac{1+\sqrt{5}}{2}$ be the Golden Number, then:

$$f_n = \frac{1}{\sqrt{5}}(\gamma^{n+1} + (-\frac{1}{\gamma})^{n+1})$$

As $n \to \infty$, the first term of f_n tends to $+\infty$ while the second tends to 0 so that $f_n = O(\gamma^{n+1})$. Based on the choice of r_0 and r_1 in I_0, $e_0 < \delta^{f_0}$ and $e_1 < \delta^{f_1}$. By induction, assuming that $e_k < \delta^{f_k}$, $\forall k \leq n-1$, then using (40), one has:

$$e_n \leq e_{n-1}e_{n-2} < \delta^{f_{n-1}+f_{n-2}} = \delta^{f_n}, \forall n \geq 2. \tag{42}$$

As $\delta < 1$, this last inequality proves the first part of the theorem, i.e. that

$$\lim_{n\to\infty} e_n = 0.$$

As for the second part of the theorem, given that:

$$|r - r_n| \leq \frac{1}{C}e_n < t_n = \frac{1}{C}\delta^{f_n},$$

then:

$$\frac{t_n}{t_{n-1}^{\gamma}} = C^{\gamma-1}\delta^{f_n-\gamma f_{n-1}}.$$

Note that

$$f_n - \gamma f_{n-1} = \frac{1}{\sqrt{5}}(\gamma^{n+1} + (-\frac{1}{\gamma})^{n+1} - \gamma^{n+1} - \gamma(-\frac{1}{\gamma})^n) = \frac{2}{\sqrt{5}}(-\frac{1}{\gamma})^{n+1}.$$

Hence $f_n - \gamma f_{n-1} \to 0$ and therefore there exist a constant K such that:

$$t_n \leq K(t_{n-1})^{\gamma}.$$

■

To illustrate the Secant method, we consider the following example.

Example 6 *Approximate the root of* $f(x) = \sin x - e^{-x}$ *up to* 10 *decimal figures, in the interval (0, 2) using the Secant's method.*

This process gives the following results:

Iteration	Iterate
0	1.0
1	1.5000000000
2	0.21271008648
3	0.77325832517
4	0.61403684201
5	0.58643504642
6	0.58855440366
7	0.58853274398
8	0.58853274398

Besides computing the initial conditions, the Secant 's method requires about 6 iterations to reach a precision $p = 10$, that is 2 more than Newton's method.

Comparisons between the convergence of both Newton and Secant methods can be further made, using the inequalities (31) and (42), as $e_n = C|r - r_n|$ satisfies respectively:

1. $e_n \leq \delta^{2^n}$ in Newton's method and
2. $e_n \leq \delta^{f_n}$ in the secant method.

with $\delta = e_0 = C|r - r_0|$. Thus

$$\frac{|r - r_n|}{|r - r_0|} = \frac{e_n}{e_0} =\leq \delta^{2^n - 1}$$

for Newton's method and

$$\frac{|r - r_n|}{|r - r_0|} = \frac{e_n}{e_0} =\leq \delta^{f_n - 1}$$

for the secant method.
In the same way that this was done for the preceding methods (Corollaries 1 and 2), one can also derive the minimum number of iterations needed theoretically to reach requested precisions using the secant method. However in this chapter, in order to confirm that Newton's method is faster, we will only consider for example the specific case of $\delta = \frac{1}{2}$, seeking the minimum n_0 for which $\frac{|r-r_n|}{|r-r_0|} \leq 2^{-p}$, (i.e a precision p in a floating-point system $\mathbb{F}(2, p, E_{\min}, E_{\max})$). Straightforward it can be shown that such n_0 satisfies:

$$2^{n_0^{(1)}} \geq 1 + p > 2^{n_0^{(1)} - 1}$$

for Newton's method and

$$f_{n_0^{(2)}} \geq 1 + p > f_{n_0^{(2)} - 1}$$

for the secant method. We summarize the results in the following table:

p	$n_0^{(1)}$	$n_0^{(2)}$
10	4	6
24 (IEEE-single)	5	8
53 (IEEE-double)	6	9

Thus although Newton's method is faster, it takes at most about 2 to 3 more iterations for the secant method to reach a same precision.

6 Exercises

The Bisection Method

1. Locate all the roots of f, then approximate each one of them up to 3 decimal figures using the Bisection method.

   (a) $f(x) = x - 2\sin x$

   (b) $f(x) = x^3 - 2\sin x$

   (c) $f(x) = e^x - x^2 + 4x + 3$

   (d) $f(x) = x^3 - 5x - x^2$

2. Show that the following equations have infinitely many roots by graphical methods. Use the Bisection method to determine the smallest positive value up to 4 decimal figures.

   (a) $\tan x = x$

   (b) $\sin x = e^{-x}$

   (c) $\cos x = e^x$

   (d) $\ln(x+1) = \tan(2x)$

3. The following functions have a unique root in the interval $[1, 2]$. Use the Bisection method to approximate that root up to 3 decimal figures. Compare the number of iterations needed to reach that precision with the predictable "theoretical" value.

   (a) $f(x) = x^3 - e^x$

   (b) $f(x) = x^2 - 4x + 4 - \ln x$

   (c) $f(x) = x^3 + 4x^2 - 10$

   (d) $f(x) = x^4 - x^3 - x - 1$

   (e) $f(x) = x^5 - x^3 + 3$

   (f) $f(x) = e^{-x} - \cos x$

   (g) $f(x) = \ln(1+x) - \frac{1}{x+1}$

4. The following functions have a unique root in the interval $[0, 1]$. Use the Bisection method to approximate that root up to 5 decimal figures. Compare the number of iterations needed to reach that precision with the predictable "theoretical" value.

   (a) $f(x) = e^{-x} - 3x$

   (b) $f(x) = e^x - 2$

   (c) $f(x) = e^{-x} - x^2$

(d) $f(x) = \cos x - x$
(e) $f(x) = \cos x - \sqrt{x}$
(f) $f(x) = e^x - 3x$
(g) $f(x) = x - 2^{-x}$
(h) $f(x) = 2x + 3\cos x - e^x$
(i) $f(x) = \sin x - x^3$

5. Prove that the function $f(x) = ln(1-x) - e^x$ has a unique negative root. Use the Bisection method to calculate the first four iterations.

6. Prove that the function $f(x) = e^x - 3x$ has a unique positive root. Use the Bisection method to calculate the first four iterations.

7. The bisection method generates a sequence of intervals $\{[a_0, b_0], [a_1, b_1], ...\}$. Prove or disprove the following estimates.

   (a) $|r - a_n| \leq 2|r - b_n|$
   (b) $|r - b_n| \leq 2^{-n}(b_0 - a_0)$
   (c) $r_{n+1} = \frac{a_n + r_n}{2}$
   (d) $r_{n+1} = \frac{b_n + r_n}{2}$
   (e) $|r - a_n| \leq 2|r - b_n|$
   (f) $|r - b_n| \leq 2^{-n}|b_0 - a_0|$

Newton's and the Secant Methods

8. Use three iterations of Newton's method to compute the root of the function $f(x) = e^{-x} - \cos x$ that is nearest to $\pi/2$

9. Use three iterations of Newton's method to compute the root of the function $f(x) = x^5 - x^3 - 3$ that is nearest to 1.

10. The polynomial $p(x) = x^4 + 2x^3 - 7x^2 + 3$ has 2 positive. Find them by Newton's method, correct to four significant figures.

11. Use Newton's method to compute $\ln 3$ up to five decimal figures.

12. Approximate $\pm\sqrt{e}$ up to 7 decimal figures using Newton's method.

13. Compute the first four iterations using Newton's method to find the negative root of the function $f(x) = x - e/x$.

14. Use Newton's method to approximate the root of the following functions up to 5 decimal figures, located in the interval $[0, 1]$. Compare the number of iterations used to reach that precision with the predictable "theoretical" value.

(a) $f(x) = e^x - 3x$

(b) $f(x) = x - 2^{-x}$

(c) $f(x) = 2x + 3\cos x - e^x$

(d) $f(x) = \sin x - x^3$

15. To approximate the reciprocal of 3, i.e. $r = \frac{1}{3}$, using Newton's method:

    (a) Define some appropriate non polynomial function that leads to an iterative formula not dividing by the iterate. Specify the restrictions on the initial condition if there are any.

    (b) Choose two different values for the initial condition to illustrate the local character of convergence of the method.

16. Based on Newton's method, approximate the reciprocal of the square root of a positive number R, i.e. $\frac{1}{\sqrt{R}}$, using first a polynomial function, and secondly a non polynomial function. Determine the necessary restrictions on the initial conditions, if there are any.

17. To approximate the negative reciprocal of the square root of 7, i.e. $r = \frac{-1}{\sqrt{7}}$, using Newton's method:

    (a) Define some appropriate non polynomial function that leads to an iterative formula not dividing by the iterate. Specify the restrictions on the initial condition if there are any.

    (b) Use Newton's method to approximate $r = \frac{-1}{\sqrt{7}}$ up to 4 decimal figures.

    item Approximate $\sqrt{2}$ up to 7 decimal figures using Newton's method.

18. The number $\sqrt{R}$ $(R > 0)$, is a zero of the functions listed below. Based on Newton's method , determine the iterative formulae for each of the functions that compute $\sqrt{R}$. Specify any necessary restriction on the choice of the initial condition, if there is any.

    (a) $a(x) = x^2 - R$

    (b) $b(x) = 1/x^2 - 1/R$

    (c) $c(x) = x - R/x$

    (d) $e(x) = 1 - R/x^2$

    (e) $g(x) = 1/x - x/R$

    (f) $h(x) = 1 - x^2/R$

19. Based on Newton's method , determine an iterative sequence that converges to π. Compute π up to 3 decimal figures.

20. Let $f(x) = x^3 - 5x + 3$.
a - Locate all the roots of f.
b - Use successively the Bisection and Newton's methods to approximate the largest root of f correct to 3 decimal places.
c- How many iterations are theoretically needed using each method, to calculate a root up to 3 decimal places ? Compare these values with the results obtained in (b).

21. To approximate the cubic root of a positive number a, i.e. $r = a^{\frac{1}{3}}$, where $1 < a \leq 2$, using Newton's method:

    (a) Define some appropriate polynomial function $f(x)$ with unique root $r = a^{\frac{1}{3}}$, then write the formula of Newton's iterative sequence $\{r_n\}$.

    (b) Assume that, for $r_0 = 2$, the sequence $\{r_n\}$ is decreasing and satisfies: $a^{\frac{1}{3}} = r < ... < r_{n+1} < r_n < r_{n-1} < ... < r_1 < r_0 = 2$.
    Prove then that: $|r_{n+1} - r| \leq (r_n - r)^2$ for all $n \geq 0$.

    (c) Prove by recurrence that: $|r - r_n| \leq |r - r_0|^{2^n}$, for all $n \geq 0$

    (d) Assuming $|r_0 - r| \leq \frac{1}{2}$. Estimate the least integer n_0 such that $|r_{n_0} - r| \leq (\frac{1}{2})^{32}$.

22. Let $p(x) = c_2x^2 + c_1x + c_0$ be a **quadratic polynomial** with one of its roots r located in an interval (a, b), with

$$\min_{a \leq x \leq b} |p'(x)| \geq d > 0$$

Using Newton's method with r_0 sufficiently close to r:
a- Show that if $r_n \in (a, b)$ then $|r_{n+1} - r| \leq C|r_n - r|^2$, where $C = \frac{|c_2|}{d}$.
b - Let $e_n = C|r - r_n|$. Show that if $r_n \in (a, b)$ then $e_{n+1} \leq e_n^2$. Give also the condition on $|r_0 - r|$ that makes $e_0 < 1$, and therefore $e_n < 1$ for all n.
c- Assume $|r_0 - r| = \frac{1}{2C}$. Show by recurrence that $e_n \leq (e_0)^{(2^n)}$, then estimate the smallest value n_p of n, so that:

$$\frac{|r_{n_p} - r|}{|r_0 - r|} \leq 2^{-p}.$$

23. Calculate an approximate value for $4^{3/4}$ using 3 steps of the secant method.

24. Use three iterations of the Secant method to approximate the smallest positive root of $f(x) = x^3 - 2x + 2$.

25. Show that the iterative formula for the secant method can also be written

$$x_{n+1} = \frac{x_{n-1}f(x_n) - x_nf(x_{n-1})}{f(x_n) - f(x_{n-1})}$$

Compare it with the standard formula (**??**. Which one is more appropriate to use in the Algorithm of the Secant method?

26. Use the Secant method to approximate the root of the following functions up to 5 decimal figures, located in the interval $[0, 1]$. Compare the number of iterations used to reach that precision with the number of iterations obtained in exercise 15.

    (a) $f(x) = e^x - 3x$

    (b) $f(x) = x - 2^{-x}$

    (c) $f(x) = -3x + 2\cos x - e^x$

    (d) $f(x) = \sin x - x^3$

References

[1] **BURDEN, R., FAIRES D.**, *Numerical Analysis*. 7th edition. Brooks, Cole. 2001

[2] **CHENEY, W., KINCAID D.**, *Numerical Mathematics and Computing*. 4th edition. Brooks, Cole 1999.

[3] **ISAACSON, E., KELLER H.B**, *Analysis of Numerical Methods*. 4th edition. Wiley 1966.