

2. Solving Nonlinear Equations

2.1 Introduction

The problem of solving the nonlinear equation $f(x)=0$ is a task that frequently arises in scientific computing, either in its raw form or as part of more complex problems. It is relatively rare to be able to obtain its roots with precision. Furthermore, in some cases, the coefficients of the equation are known only approximately, and consequently, the problem of precisely determining the roots itself loses its meaning. This is why methods for finding approximate roots of an equation and estimating the degree of their accuracy have become particularly important.

Practical Example:

Among the nonlinear problems described in physics and engineering is the problem of the floating sphere:

Consider a solid sphere of radius r floating on the surface of water. Let h denote the depth of the submerged portion of the sphere. According to Archimedes' principle, the buoyant force exerted on the sphere is equal to the weight of the displaced volume of water. At static equilibrium, this buoyant force exactly balances the weight of the sphere.

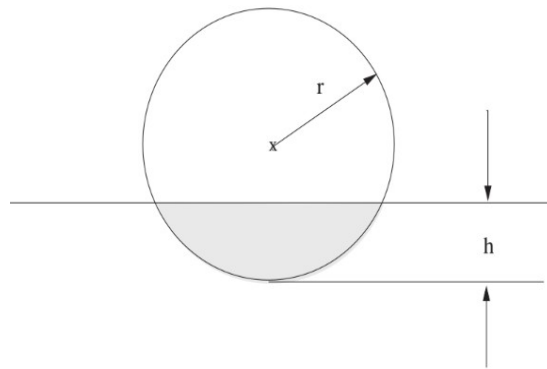


Fig.1 : Archimedes' problem

$$V_s = \frac{4}{3} \pi r^3$$

2. Solving Nonlinear Equations

Let V_s be the volume of the sphere, and let V_{water} be the volume of water it displaces when partially submerged.

$$V_{water} = \frac{\pi}{3} h^2 (3r - h)$$

At static equilibrium, the weight of the sphere is balanced by the buoyant force

$$\rho_s g V_s = \rho_{water} g V_{water}$$

which leads to the following equation in terms of h :

$$h^3 - 3rh^2 + 4 \frac{\rho_s}{\rho_{water}} r^3 = 0$$

2.2 Root of an equation:

Let $f(x)$ be a function defined and continuous on a finite or infinite interval $a < x < b$. Any value α that sets the function $f(x)$ to zero (i.e., $f(\alpha) = 0$) is called a root of the equation $f(x) = 0$ or a zero of f .

Order of a Root

A root of the equation $f(x) = 0$ of order n is called α if it satisfies:

$$f(x) = (x - \alpha)^n g(x) \quad (2.1)$$

Where $g(x)$ is a continuous function and $g(\alpha) \neq 0$. A real number α is called a **simple root** of $f(x)$ if it is of order 1; otherwise, α is said to be a **multiple root**.

If the root α is of odd order, then the function changes sign in the vicinity of α . On the other hand, if the order is even, the graph of $f(x)$ is tangent to the x -axis in the vicinity of the point α .

Theorem 2.1 : If $\alpha \in [a, b]$ is a root of order n of $f(x)$, then:

$$f(\alpha) = f'(\alpha) = f''(\alpha) = \dots = f^{[n-1]}(\alpha) = 0, f^{[n]}(\alpha) \neq 0$$

Theorem 2.2 : If $f \in C^1(I)$ has a root of order 2, $\alpha \in [a, b]$, and if f is continuous and differentiable on $[a, b]$, then α is a simple root of $g(x)$:

$$g(x) = \frac{f(x)}{f'(x)}$$

2. Solving Nonlinear Equations

Existence and Uniqueness of a Solution:

To confirm the existence of a root of the equation $f(x)=0$, we use the Intermediate Value Theorem or Bolzano's Theorem:

The mathematician **Bernard Bolzano (1781–1848)**, a philosopher, priest, and mathematician from the Kingdom of Bohemia, played an important role in the effort to provide mathematical analysis with more rigorous foundations. In **1817**, he published in Prague a paper entitled:

“Purely Analytic Proof of the Theorem that Between Any Two Values Giving Results of Opposite Signs, There Lies at Least One Real Root of the Equation.”

Bolzano sought precisely to avoid basing an analytical proof on geometric intuition. His objective was to construct a **purely analytical proof**, based on the properties of numbers and sequences. His 1817 paper therefore represents an important milestone in the **rigorization of mathematical analysis**.

Augustin-Louis Cauchy also formulated and proved this result in his works on analysis, around **1821**. The modern formulation of the **Intermediate Value Theorem** is therefore generally associated with the developments of Bolzano and Cauchy.

Theorem 2.3 (Bolzano's Theorem (1781–1848)): *If a continuous function $f(x)$ takes values of opposite signs at the endpoints of the interval $[a, b]$, $f(a) \times f(b) < 0$, then this interval contains at least one root of the equation $f(x) = 0$.*

This fact implies the existence of $\alpha \in [a, b]$ such that $f(\alpha) = 0$

2. Solving Nonlinear Equations

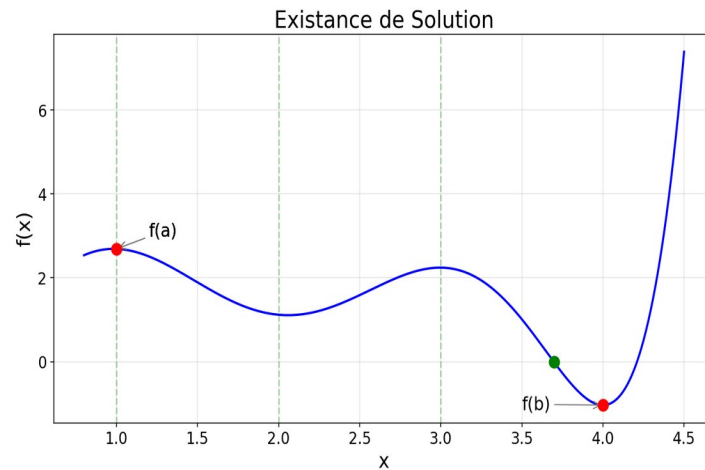


Fig.2 : Existence of Solutions

Example: Consider the function $f(x) = \cos(x) - x$. Check whether $f(x)$ has a zero in the interval $[0, 2.0]$.

Solution:

$$f(0.0) = 1.000, \quad f(2.0) = -1.0006$$

$f(x)$ is continuous on the interval $[0, 2.0]$, and since $f(0) \times f(2.0) < 0.00$, there exists a $\alpha \in [0.0, 2.0]$ such that $f(\alpha) = 0$.

To confirm the uniqueness of the root of the equation $f(x) = 0$, we use the following theorem:

Theorem 2.4 : If a continuous function $f(x)$ is monotonic on the interval $[a, b]$ and satisfies Bolzano's theorem, then the solution to $f(x) = 0$ is unique.

The approximate search for the roots of an equation is generally carried out in two steps:

1_ Root separation: This involves establishing the tightest possible segments $[a, b]$ possible intervals containing one and only one root.

2_ Improving the accuracy or refining the approximate roots, that is, achieving the required precision.

2.3 BISECTION Method (Dichotomy)

In computer science, dichotomy (from the Greek for “to cut in half”) is an iterative or recursive search process in which, at each step, the search space is restricted to one of two parts.

If f is a continuous function on the interval $[a, b]$ and changes sign that is, $f(a) \times f(b) < 0$ and to find the solution of the equation

$$f(x) = 0 \tag{2.2}$$

We divide this interval into two parts, $[a, x_1]$ and $[x_1, b]$, where:

2. Solving Nonlinear Equations

$$x_1 = \frac{a+b}{2} \quad (2.3)$$

if $f(x_1) = 0$, then the solution to (1) is x_1 ; otherwise, we search for the solution in the other two intervals by applying

Theorem 2.5 : If $f(a) \times f(x_1) < 0$, then the solution we are seeking lies in this interval $[a, x_1]$, denoted by $[a_1, b_1]$ where $a_1 = a$ and $b_1 = x_1$. Otherwise, the solution lies in the other interval $[x_1, b]$, denoted by $[a_1, b_1]$ $a_1 = x_1$ and $b_1 = b$, and we continue the search process until we obtain an interval $[a_n, b_n]$ for which

$$|b_n - a_n| < \varepsilon \quad (2.4)$$

Where $f(x_n) < \xi$ such that $x_n = \frac{a_n + b_n}{2}$, with ε and ξ being the required precisions for the solution sought

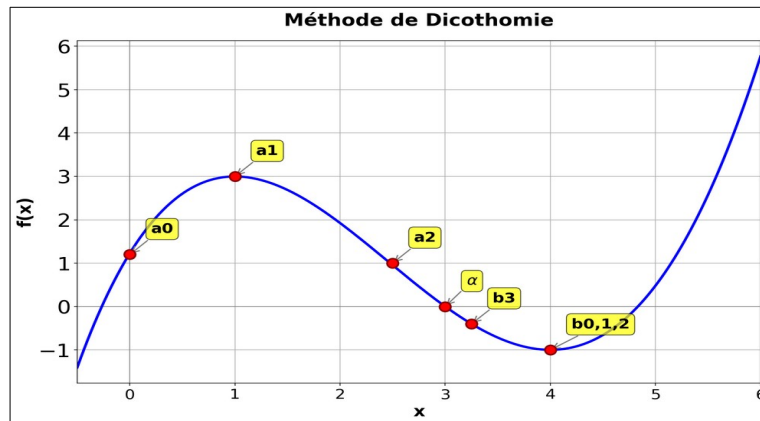


fig.3 : Bisection Method

2. Solving Nonlinear Equations

Algorithm:

```

Start
Read a, b,
Do while  $|b - a| > \varepsilon$ 
 $x = (a + b)/2$ 
If  $f(x) = 0$ , stop process
end if
If  $f(a).f(x) < 0$        $b = x$ 
else                   $a = x$ 
end if
End while
    
```

Example: Solve the equation $e^x + 2x = 0$ for $\varepsilon = 0.001$ in the interval $[-1.0, 1.0]$

Solution:

a	b	x	$f(a)$	$f(b)$	$f(x)$
-1.000000	0.000000	-0.500000	-1.6321206	1.0000000	-0.3934693
-0.500000	0.000000	-0.250000	-0.3934693	1.0000000	0.2788008
-0.500000	-0.250000	-0.375000	-0.3934693	0.2788008	-0.0627107
-0.375000	-0.250000	-0.312500	-0.0627107	0.2788008	0.1066156
-0.375000	-0.312500	-0.343750	-0.0627107	0.1066156	0.0216062
-0.375000	-0.343750	-0.359375	-0.0627107	0.0216062	-0.0206375
-0.359375	-0.343750	-0.3515625	-0.0206375	0.0216062	0.0004629
-0.359375	-0.3515625	-0.3554688	-0.0206375	0.0004629	-0.0100927

Table1 : Solving a nonlinear equation using the bisection method

After a certain number of iterations, the solution is obtained: $x = -0.3517337$

Number of iterations:

In most calculations, the number of iterations required to determine the approximate solution to a certain precision ε determines the value of the chosen method. For the *bisection* method, the number of iterations is given by:

$$N \geq \log_2 \left(\frac{|a - b|}{\varepsilon} \right) \quad (2.5)$$

Exercise: Prove the preceding formula.

2. Solving Nonlinear Equations

Python code for the bisection method:

Running the DICO.py code (Appendix 1) yields the following result:

```
0 0.00000000 1.00000000 0.50000000 -0.17563936
1 0.50000000 1.00000000 0.75000000 0.58775001
2 0.50000000 0.75000000 0.62500000 0.16765372
3 0.50000000 0.62500000 0.56250000 -0.01278176
4 0.56250000 0.62500000 0.59375000 0.07514236
5 0.56250000 0.59375000 0.57812500 0.03061924
6 0.56250000 0.57812500 0.57031250 0.00878000
7 0.56250000 0.57031250 0.56640625 -0.00203538
8 0.56640625 0.57031250 0.56835938 0.00336366
9 0.56640625 0.56835938 0.56738281 0.00066198
10 0.56640625 0.56738281 0.56689453 -0.00068724
11 0.56689453 0.56738281 0.56713867 -0.00001276
12 0.56713867 0.56738281 0.56726074 0.00032458
13 0.56713867 0.56726074 0.56719971 0.00015590
num iti= 13
x = 0.56719970703125
```

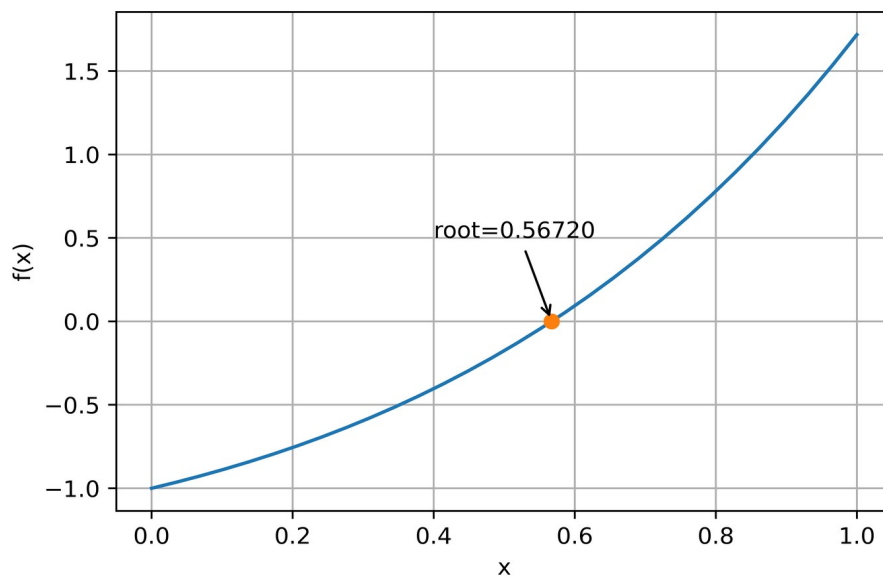


Fig. 4: curve $f(x)$

2. Solving Nonlinear Equations

2.4 False Position Method (Regula Falsi)

In this method, finding the solution to $f(x)=0$ within the interval $[a, b]$ follows the same process as the bisection method, but instead of dividing the segment $[a, b]$ into two equal intervals, it makes more sense to divide it in the ratio $-f(a)/f(b)$. This yields the approximate value of the root:

$$x_1 = a + h_1, h_1 = \frac{-f(a)}{f(b) - f(a)}(b - a) \quad (2.6)$$

Or, more simply

$$x_1 = \frac{af(b) - bf(a)}{f(b) - f(a)} \quad (2.7)$$

Graphically, this amounts to finding the point of intersection between the line passing through the points $(a, f(a))$ and $(b, f(b))$ and the x-axis: (Figure .5):

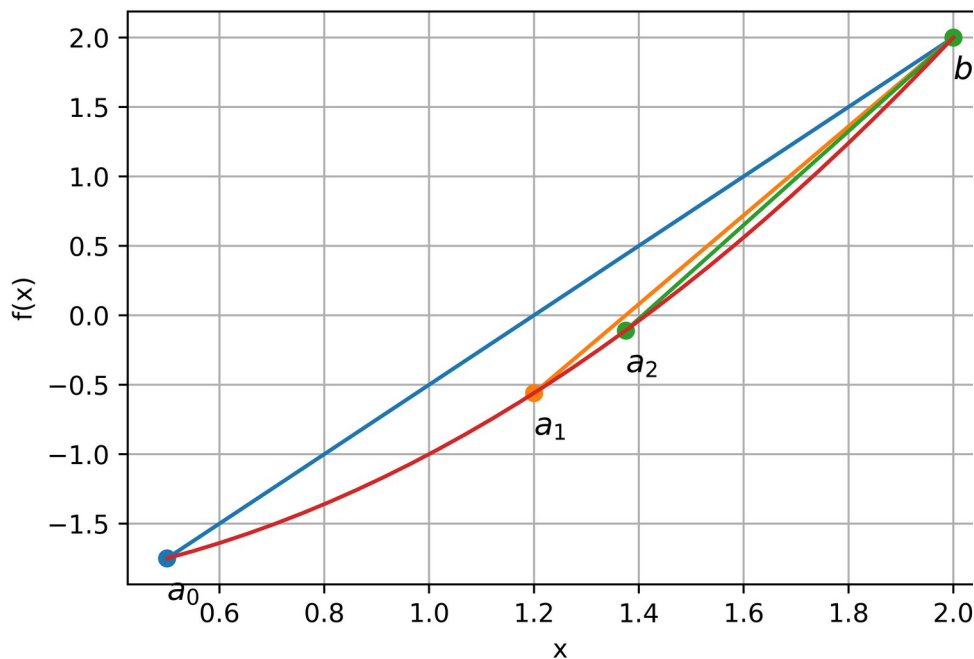


Fig.5 : Regula-falsi method

2. Solving Nonlinear Equations

$$y(x) = f(a) + \frac{f(b) - f(a)}{b - a}(x - a) \quad \text{when } y=0 \quad x = \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)}$$

And we repeat the same search process as in the bisection method, that is:

Algorithm:

```

Start
Read a, b, ε
Do
  x = (a · f(b) - b · f(a)) / (f(b) - f(a))
  If f(x) = 0, stop the process
  End if
  If f(a)f(x) < 0      b = x
  Else                a = x
  End if
as long as |f(x)| > ε
  
```

Example: Find $\alpha = \sqrt{2}$ using the false position method and compare the result (the number of operations) with that obtained using the bisection method ($\varepsilon = 0.001$).

Solution: To find $\alpha = \sqrt{2}$, we have $x = \sqrt{2}$, so $x^2 = 2$, which implies $x^2 - 2.0 = 0$. Let's take $f(x) = x^2 - 2$ in the interval $[1.0, 2.0]$ and apply the Regula-Falsi method:

a	b	x	f(a)	f(b)	f(x)
1.0000000	2.0000000	1.3333333	-1.0000000	2.0000000	-0.2222222
1.3333333	2.0000000	1.4000000	-0.2222222	2.0000000	-0.0400000
1.4000000	2.0000000	1.4117647	-0.0400000	2.0000000	-0.0069204
1.4117647	2.0000000	1.4137931	-0.0069204	2.0000000	-0.0011891
1.4137931	2.0000000	1.4141414	-0.0011891	2.0000000	-0.0002041
1.4141414	2.0000000	1.4142012	-0.0002041	2.0000000	-0.0000350
1.4142012	2.0000000	1.4142114	-0.0000350	2.0000000	-0.0000060
1.4142114	2.0000000	1.4142132	-0.0000060	2.0000000	-0.0000010
1.4142132	2.0000000	1.4142135	-0.0000010	2.0000000	-0.0000002
1.4142135	2.0000000	1.4142136	-0.0000002	2.0000000	0.0000000
1.4142136	2.0000000	1.4142136	0.0000000	2.0000000	0.0000000

Table2 : Solution using the false position method

The solution is $\alpha = 1.4142136$

Compare the number of operations performed to find the solution to this equation with those of the bisection method.

2. Solving Nonlinear Equations

Python code for Regula-Falsi: $\sqrt{2}$, which is the root of the equation $x^2 - 2.0 = 0$, for $x \in [1.0, 2.0]$ and $\epsilon = 10^{-4}$

The Python code REGULA.py:

```
0 1.00000000 2.00000000 1.33333333 -0.22222222
1 1.33333333 2.00000000 1.40000000 -0.04000000
2 1.40000000 2.00000000 1.41176471 -0.00692042
3 1.41176471 2.00000000 1.41379310 -0.00118906
4 1.41379310 2.00000000 1.41414141 -0.00020406
5 1.41414141 2.00000000 1.41420118 -0.00003501
num iti= 5
x = 1.4142011834319526
```

2.5 Secant Method:

It is possible to use the false position algorithm to construct a sequence of solutions without checking the solution interval. In this case, the method is called the secant method.

The method consists of taking the interval bounds as the first two solutions $[a, b]$:

$x_0 = a$ $x_1 = b$ for $i > 1$

$$x_{n+1} = x_n - f(x_n) \frac{(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})} \quad (2.8)$$

the algorithm becomes:

```
Start
Read a, b, ε
x0 = a
x1 = b
n = 1
Do |xn - xn-1| > ε
  xn+1 = xn - f(xn) (xn - xn-1) / (f(xn) - f(xn-1))
  n = n + 1
end fair
end
```

Example:

Find the solution to the equation $f(x) = e^x - x - 2.0$ using the secant method in the interval $[1.0, 2.0]$:

Solution: Let $x_1 = 1.0$, $x_2 = 2.0$ and use the following algorithm:

2. Solving Nonlinear Equations

$$x_{n+1} = x_n - f(x_n)(x_n - x_{n-1}) / (f(x_n) - f(x_{n-1}))$$

Then:

k	x_k	$f(x_k)$
0	1,000	-0.28171817
1	2,000	3.38905610
2	1.07674625	-0.14163237
3	1.11378226	-0.06792539
4	1.14791316	0.00369599
5	1.14615186	-0.00008878
6	1.14619317	-0.00000011

Table3 : Table Example

Python code SECANTE.py:

```
0 0.00000000 -1.00000000
1 2.00000000 3.38905610
2 0.45567884 -0.87843512
3 0.77356724 -0.60608282
4 1.48098443 0.91628792
5 1.05520258 -0.18264556
6 1.12596862 -0.04276678
7 1.14760474 0.00303253
8 1.14617214 -0.00004525
9 1.14619320 -0.00000005
iti = 9 x = 1.1461931988205012
```

2. Solving Nonlinear Equations

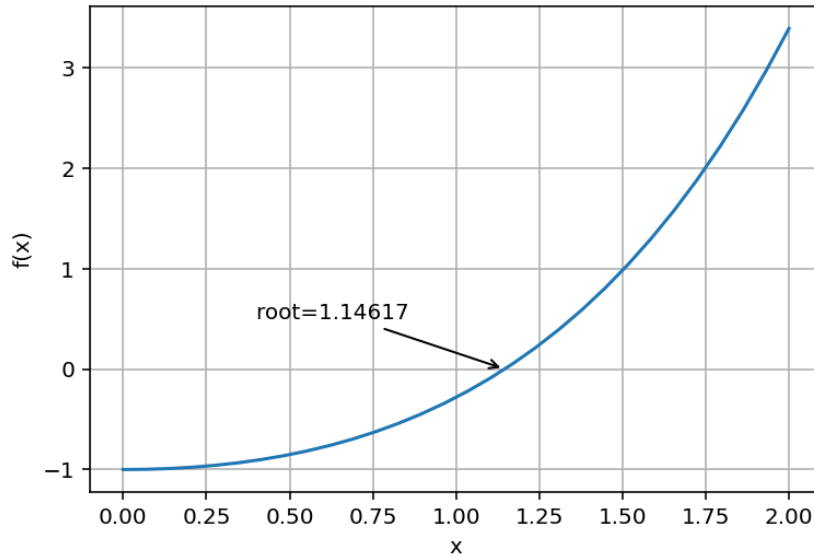


Fig.6 : Solution Using the Secant Method

Order of convergence:

If we approximate a solution α of an equation of the form $f(x)=0$ using a recursive sequence x_n and this sequence satisfies the condition:

$$\exists C \in [a, b] \quad \frac{|x_{n+1} - \alpha|}{|x_n - \alpha|^p} \leq C < 1 \quad \forall n > n_0 \quad (2.9)$$

then the recursive sequence converges to α with order p .

Where $n \geq 0$ is a suitable integer. In such cases, the method is said to be of order p . Note that if p equals 1, for x_n to converge to α , C must be less than 1 ($C < 1$); in this case, C is the convergence factor. If $p=1$, the sequence converges linearly; if $p=2$, we speak of quadratic convergence.

The bisection method converges globally, and the value of C cannot be estimated; it is p . The Regula-Falsi method is of order 1, as is the secant method: $p \approx 1.618$ (superlinear) it is possible to determine experimentally the order of convergence of a method by studying the following sequence:

$$P_k = \frac{\log|x^k - \alpha|}{\log|x^{k-1} - \alpha|} \quad P = \lim_{k \rightarrow \infty} P_k, \quad (2.10)$$

2. Solving Nonlinear Equations

Example: For the example of calculating $\sqrt{2}$ using the Regula-Falsi method, we will study the sequence P_k to see:

i	x	P_k
0	1.200000	-21.417640
1	1.375000	0.504688
2	1.407407	0.910187
3	1.413043	0.984098
4	1.414013	0.997256
5	1.414179	0.999529
6	1.414208	0.999919

Tab.4 : Example Table

2.6 Newton-Raphson method:

Newton's method was discovered by Isaac Newton and published in *Method of Fluxions* in 1736. Although the method was described by Joseph Raphson (1648–1715) in “Analysis Aequationum” in 1690, the relevant sections of “Method of Fluxions” had already been written in 1671.

Newton's method is based on the concept of the derivative. If x_0 is an approximation of the solution to $f(x) = 0$ on the interval $[a, b]$, we correct this solution by setting $x_1 = x_0 + h$, where h is a small value. Using the Taylor series expansion:

$$f(x_1) = f(x_0 + h) \approx f(x_0) + h \cdot f'(x_0)$$

If $f(x_1) = 0$, then $h = -\frac{f(x_0)}{f'(x_0)}$, so

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

And for a certain step size n :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (2.11)$$

We can see that when x_n is a solution, then $f(x_n) \rightarrow 0$, so $x_n \rightarrow x_{n+1}$.

2. Solving Nonlinear Equations

Graphical Interpretation

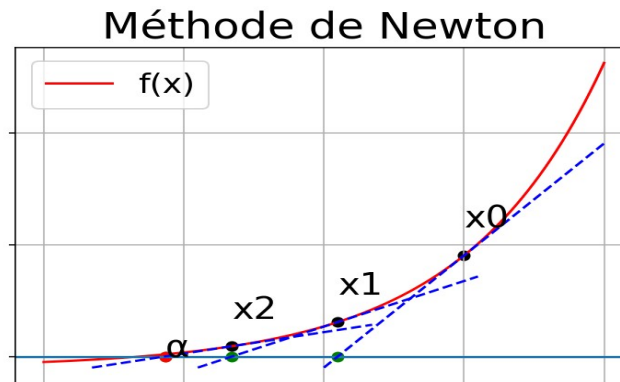


Fig.7 : Graphical interpretation of Newton's method

We start with an initial solution at the point x_0 and find the intersection of the tangent line to the function $f(x)$ at that point with the x-axis, denoted by x_1 , which will be the new solution; we continue this process until we obtain a good approximation.

Algorithm

```
Start
Read  $x_0, \varepsilon$ 
 $n=0$ 
Do
 $x_{n+1} = x_n - f(x_n)/f'(x_n)$ 
 $n = n + 1$ 
as long as  $|x_n - x_{n-1}| > \varepsilon$ 
end
```

Example: Repeat the previous exercise using Newton's method, where $x_0 = 2.0$.

Solution:

2. Solving Nonlinear Equations

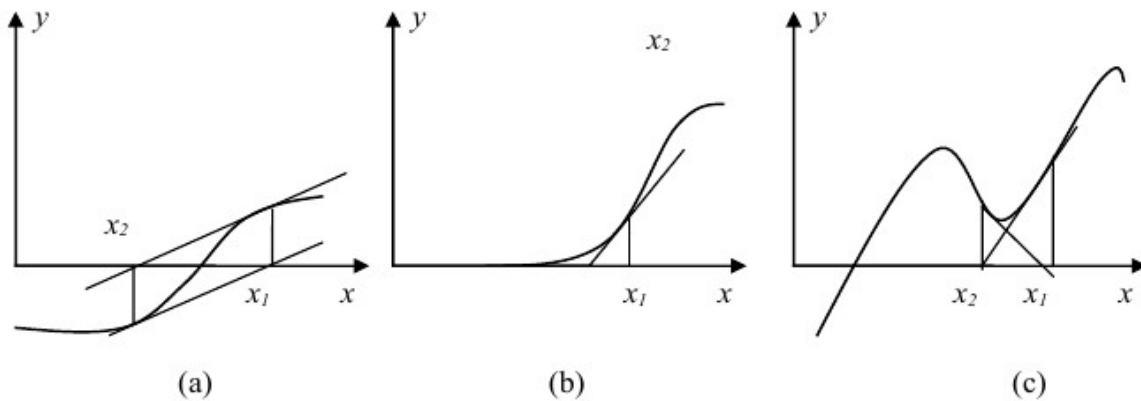
i	x_i	$f'(x_i)$	$f(x_i)$
0	2.0000000	4.0000000	2.0000000
1	1.5000000	3.0000000	0.2500000
2	1.4166666	2.8333333	0.0069444
3	1.4142156	2.8284313	0.0000060
4	1.4142135	2.8284271	0.0000000

Table5 : Solution using Newton's method

Note:

Newton's method is very practical in terms of convergence speed, but there are situations where divergence is striking; this is due to:

- 1)The existence of an inflection point in the interval under consideration (Figure 2.7.a).
- 2)The existence of a multiple root of $f'(x)=0$ (Figure 2.7.b).
- 3)The presence of local minima (Figure 2.7.c).



Fig

.8 : Divergence of Newton's method

Although the method is very efficient, certain practical considerations must be taken into account. First and foremost, Newton's method requires that the derivative be actually calculated. In cases where the derivative is only estimated by taking the slope between two points of the function, the method is called *the secant method*, which is less efficient (order 1.68) and inferior to other algorithms. Furthermore, if the initial value is too far from the true zero, Newton's method can enter an infinite loop without producing an improved approximation. Because of this, any practical implementation of Newton's method must include code to control the number of iterations.

2. Solving Nonlinear Equations

Order of Convergence of Newton's Method

Let $\alpha \in [a, b]$ be a solution to the equation $f(x) = 0$ using a finite expansion

$$f(x) = f(x_n) + f'(x_n)(x - x_n) + \frac{1}{2}f''(\eta)(x - x_n)^2$$

or η is an unknown value between x et x_n if $x = \alpha$ (the solution), then $f(x) = f(\alpha) = 0$

$$0 = f(\alpha) = f(x_n) + f'(x_n)(\alpha - x_n) + \frac{1}{2}f''(\eta)(\alpha - x_n)^2$$

if we take the limit as $f'(x_n)$ and rearrange, then:

$$\alpha - \left(x_n - \frac{f(x_n)}{f'(x_n)} \right) = \frac{-f''(\eta)}{2f'(x_n)}(\alpha - x_n)^2$$

by the Newton method formula

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \text{ then:}$$

$$x_{n+1} - \alpha = \frac{f''(\eta)}{2f'(x_n)}(x_n - \alpha)^2$$

$$\text{is if we take } C = \left| \frac{f''(\eta)}{2f'(x_n)} \right|$$

then:

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - \alpha|}{|(x_n - \alpha)|^2} = C \quad (2.12)$$

and for large n , $f''(\eta) \approx f''(x_n)$; therefore, the convergence of Newton's method is quadratic if

$$\frac{f''(x_n)}{2f'(x_n)} < 1 \text{ or } \left| \frac{\max f''(x)}{2 \min f'(x)} \right| < 1$$

Choosing the initial solution for Newton's method:

Graphically, if we choose an initial solution $x_0 = a$ such that $f(x_0) \cdot f''(x_0) < 0$, the tangent to the curve $y = f(x)$ at point $A(a, f(a))$ generates a point x_1 outside the interval $[a, b]$. For this reason, to ensure the method converges correctly, it is preferable to use:

$$f(x_0) \cdot f''(x_0) > 0 \quad (2.13)$$

Convergence conditions for Newton's method:

a) Local Convergence Conditions:

1. $f(a) \cdot f(b) < 0$: The solution exists in the interval I near the solution x_0

2. $f \in C^2[a, b]$: $f(x)$ and its derivative $f'(x)$ are continuous on the interval I in the vicinity of the solution.

2. Solving Nonlinear Equations

3. $\forall x \in I, f'(x) \neq 0$: The derivative must not vanish in the interval I that contains the solution.

4. $\forall x \in I, f''(x) \neq 0$: The second derivative must maintain the same sign on the interval I.

5. $f(x_0) \cdot f''(x_0) > 0$.

b) *Global Convergence Conditions*: $f(x_0) \cdot f''(x_0) < 0$.

1. Conditions 1, 2, 3, and 4 of the local conditions.

2. $\left| \frac{f(c)}{f'(c)} \right| \leq |b-a|, \begin{cases} c=a, |f'(a)| \leq |f'(b)| \\ c=b, |f'(a)| > |f'(b)| \end{cases}$.

c) *Kantorovich's Theorem* (Convergence Condition for Newton's Method):

1. $f \in C^2[a, b]$

2. $f'(x_0) \neq 0$

3. $\max_{x \in [a, b]} \left| f''(x) \right| \left| \frac{f(x_0)}{f'(x_0)^2} \right| < \frac{1}{2}$

Python code NEWTON-RAPHSON.py: for $f(x) = xe^x - 1$, $f'(x) = (1+x)e^x$, $x_0 = 0.0$

```
1 0.00000000 -1.00000000
2 1.00000000 1.71828183
3 0.68393972 0.35534255
4 0.57745448 0.02873389
5 0.56722974 0.00023889
iti = 5 x = 0.5671432965302959
```

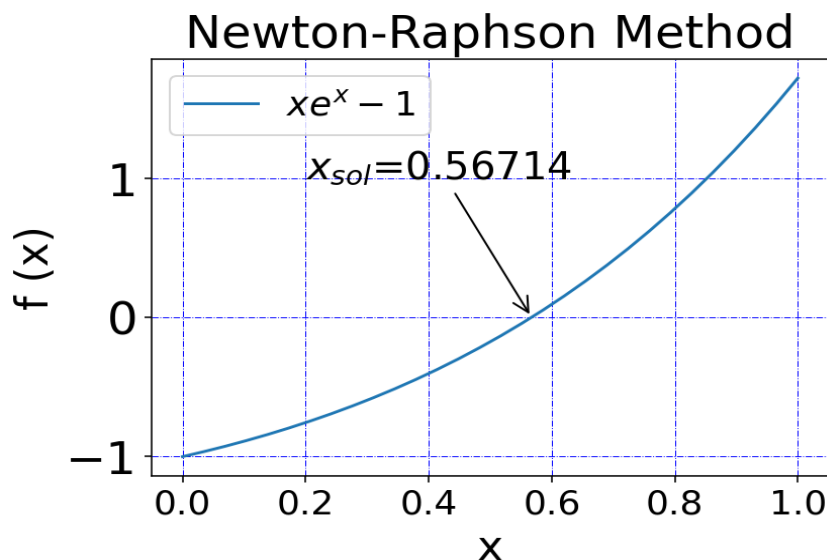


Fig.9 : Example of Newton's method

2. Solving Nonlinear Equations

2.7 Newton's method for multiple roots:

When a root is multiple, it is possible to improve the solution using two different approaches: the first is to modify the algorithm in (2.11) to:

$$x_{n+1} = x_n - m \frac{f(x_n)}{f'(x_n)} \quad (2.14)$$

where m is the order of multiplicity of the root; in the case of the following equation, $m = 2$, so:

Example:

Consider the following equation

$(x \log(x) - 1.0)^2 = 0$ admits a double root near the point $x = 1.7$ to efficiently determine this solution with an accuracy of the order of $\varepsilon = 10^{-6}$

if we use the classical Newton method, the solution is obtained after 17 iterations; if we use the algorithm from Equation 2.14 (the `Newton_Multy.py` code), the difference is clear:

$m=1$

```
0 2.0
1 1.885924163724462
2 1.825821977828081
3 1.7948612258306231
4 1.779130520335997
5 1.7711993057456277
6 1.767216792595883
7 1.7652212524360191
8 1.7642224041843864
9 1.7637227095975687
10 1.7634727945736035
11 1.7633478201145343
12 1.76328532864642
13 1.7632540818524918
14 1.7632384581905316
15 1.763230646293299
16 1.7632267403281192
17 1.7632247873413882
N_iteration = 17
```

$m=2$

```
0 2.0
1 1.7718483274489238
2 1.7632362113366402
3 1.7632228343842757
N_iteration = 3
```

2. Solving Nonlinear Equations

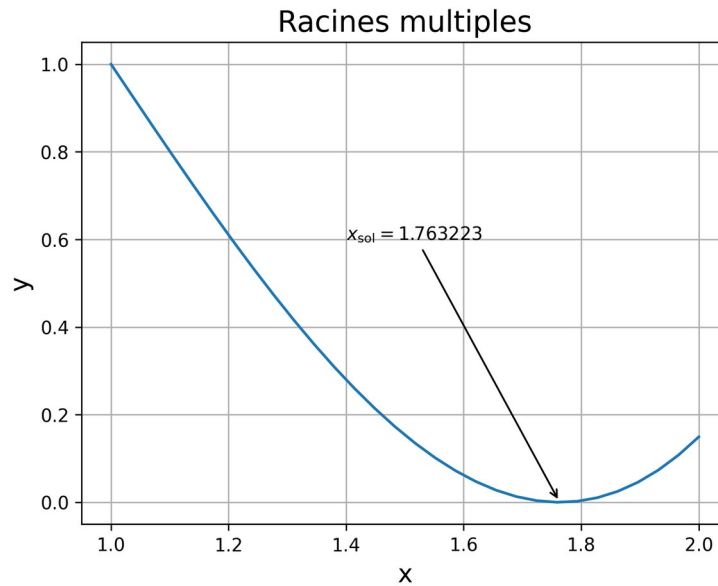


Fig. 10: Multiple roots

2.8 Successive Approximation Method

The successive approximation method is one of the oldest iterative methods. It is a very useful method for determining the roots of equations of the form $f(x)=0$, and even for systems of nonlinear equations. It is based on the following procedure:

Given the equation $f(x) = 0$, we try to rewrite it in the form $g(x) = x$, then follow this iterative procedure: starting from an initial issue $x_{(0)}$, we substitute it into $g(x)$ to obtain a new solution $x_1 = g(x_0)$, and we repeat this process until we obtain an acceptable solution. Thus, we construct a numerical sequence of values (x_n) whose values converge to the solution of the equation $f(x) = 0$.

The iterative form:

$$\begin{array}{l} x_{n+1} = g(x_n) \\ x_0 \text{ given} \end{array} \quad (2.15)$$

The iteration process stops when $|x_{n+1} - x_n| < \varepsilon$ or $f(x_n) < \zeta$.

Example:

Let's consider the equation $x^2 - 2x - 3 = 0$ (which has roots -1 and 3).

We can rewrite it as $x = g_1(x) = \sqrt{2x+3}$, and for $x_0 = 4.0$, we obtain the following results:

2. Solving Nonlinear Equations

n	x_n
0	4
1	3.31662
2	3.10375
3	3.03439
4	3.0114

Table6 : Solution using the successive approximation method.

Note that the values of x_n converge to 3.

It can be verified that if we change the form of the same equation $x = g_2(x) = \frac{1}{2}(x^2 - 3.0)$ the previous example, we can obtain a divergence.

Fixed-Point Theorem:

The contraction:

Let $g(x)$ be a function defined on the interval $[a, b]$. If there exists a real number $0 < k < 1$ such that for $x_1, x_2 \in [a, b]$:

$$|g(x_2) - g(x_1)| \leq k |x_2 - x_1| \quad (2.16)$$

then the function $g(x)$ is said to be contracting, and

Let $g \in [a, b]$ and for $x \in [a, b]$, assuming that g' exists, if there exists a real number $0 < k < 1$

$$|g'(x)| < k \quad \text{pour tout } x \in [a, b]$$

for a given x_0 belonging to $[a, b]$, the sequence

$$x_{n+1} = g(x_n) \quad n \geq 1$$

converges to a unique fixed point $\alpha \in [a, b]$

Proof:

Let α be a solution to the equation $f(x) = 0$, so $\alpha = g(\alpha)$

$$|x_n - \alpha| = |g(x_n) - g(\alpha)| = |g'(\xi_n)| |x_{n-1} - \alpha| \leq k |x_{n-1} - \alpha| \quad \xi_n \in [a, b]$$

applying this intuitive inequality, we obtain:

$$|x_n - \alpha| \leq k |x_{n-1} - \alpha| \leq k^2 |x_{n-2} - \alpha| \leq k^n |x_0 - \alpha| \quad (2.17)$$

2. Solving Nonlinear Equations

$k < 1$ alors $\lim_{n \rightarrow \infty} k^n = 0$

$$\lim_{n \rightarrow \infty} |x_n - \alpha| \leq \lim_{n \rightarrow \infty} k^n |x_0 - \alpha| = 0$$

Therefore, x_n converges to α

Theorem: If $g(x)$ and $g'(x)$ are two continuous functions on the interval containing the root x , and if $|g'(x)| < 1$ for all points in this interval, then $x_n = g(x_{n-1})$ is an iterative scheme that converges to the solution α of the equation $x = g(x)$.

This condition is sufficient but not necessary.

Property: (Ostrowski's theorem) Let α be a fixed point of a function g that is continuous and differentiable in a neighborhood of α . If $|g'(\alpha)| < 1$, then there exists a $\delta > 0$ such that the sequence $\{x_n\}$ converges to α for every x_0 such that $|x_0 - \alpha| < \delta$.

Order of Convergence:

In the case of the fixed-point method, the order of convergence is given by:

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - \alpha|}{|x_n - \alpha|} = |g'(\alpha)| \quad (2.18)$$

We can use the global approximation $|g'(x_n)| < \max_{x \in [a, b]} |g'(x)|$ to confirm convergence.

If the function $g(x)$ is twice bounded and differentiable, and $g'(x) = 0$ et $g''(x) \neq 0$:
then: the fixed-point method converges with order 2:

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - \alpha|}{|x_n - \alpha|^2} = \frac{1}{2} |g''(\alpha)| \quad (2.19)$$

In the general case:

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - \alpha|}{|x_n - \alpha|} < C \quad (2.20)$$

Or $C = \max_{x \in [a, b]} |g'(x)|$

Algorithm:

```

Start
Read x0
n = 0
Do
  xn+1 = g(xn)
  n = n + 1
as long as |xn - xn-1| > ε
end
  
```

2. Solving Nonlinear Equations

Graphical Interpretation:

Geometrically, this method can be explained as follows:

In the xoy plane, we construct the curve $y = g(x)$ and the line $y = x$; any root of the equation $f(x) = 0$ is the point of intersection of the two curves. Starting from a point x_0 , we construct the polygonal line (staircase, Figure II.6.(a); spiral, Figure II.6.(b))

As the method converges, the horizontal segments of the polygon narrow until they converge to a single point, which is none other than the solution to the equation $f(x) = 0$. For a spiral-shaped solution, $g'(x)$ is negative; otherwise, the staircase shape is obtained.

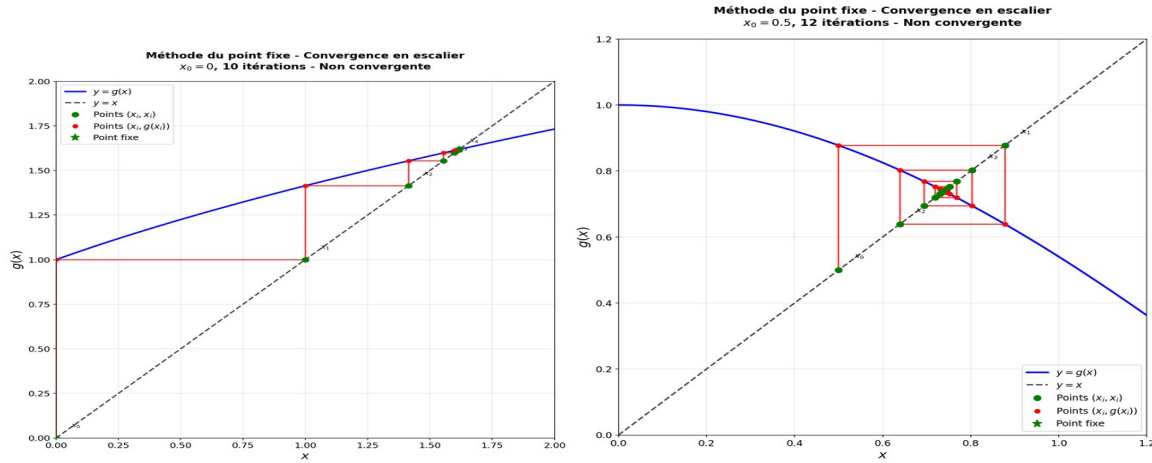


Fig.11 : Convergence of the fixed-point method

Number of iterations of the fixed-point method:

In the fixed-point method, if α is a root, then $\alpha = g(\alpha)$; therefore, it can be shown that:

$$|x_n - \alpha| \leq \frac{k^n}{1-k} |x_1 - x_0| \quad (2.21)$$

where

$$k = \max_{[a,b]} (|g'(x)|)$$

The number of iterations of the fixed-point method is given by the following formula:

if $|x_n - \alpha| \leq \varepsilon$

then

2. Solving Nonlinear Equations

$$n \geq \frac{\ln(\varepsilon) + \ln(1-k) - \ln|x_1 - x_0|}{\ln(k)} \quad (2.22)$$

Proof:

Calculating the number of iterations:

we have

$$|x_n - \alpha| \leq k|x_{n-1} - \alpha| \leq k^2|x_{n-2} - \alpha| \leq k^n|x_0 - \alpha|$$

with α est la solution et $k = \max |g'(x)|$

we have:

$$|x_0 - \alpha| = |x_0 + x_1 - x_1 - \alpha|$$

$$\begin{aligned} |x_0 - \alpha| &\leq |x_0 - x_1| + |x_1 - \alpha| \\ |x_0 - \alpha| &\leq |x_0 - x_1| + k|x_0 - \alpha| \\ |x_0 - \alpha| - k|x_0 - \alpha| &\leq |x_0 - x_1| \end{aligned}$$

$|x_0 - \alpha|(1-k) \leq |x_0 - x_1|$ so $|x_0 - \alpha| \leq \frac{1}{(1-k)}|x_0 - x_1|$, which is the same as:

$$|x_0 - \alpha| \leq \frac{1}{(1-k)}|x_1 - x_0|$$

so:

$$|x_n - \alpha| \leq k|x_{n-1} - \alpha| \leq k^2|x_{n-2} - \alpha| \leq k^n|x_0 - \alpha| \leq \frac{k^n}{(1-k)}|x_1 - x_0|$$

finally:

$$|x_n - \alpha| \leq \frac{k^n}{(1-k)}|x_1 - x_0|$$

$$|x_n - \alpha| \leq \frac{k^n}{(1-k)}|x_1 - x_0| \quad (2.23)$$

$|x_n - \alpha| \leq \frac{k^n}{(1-k)}|x_1 - x_0| \leq \varepsilon$ so: $k^n \leq \frac{(1-k)\varepsilon}{|x_1 - x_0|}$ and when we take the natural logarithm:

$$n \geq \frac{\ln((1-k)\varepsilon) - \ln(|x_1 - x_0|)}{\ln(k)}$$

2. Solving Nonlinear Equations

Convergence Conditions for the Fixed-Point Method:

To ensure convergence of the fixed-point method, the following conditions must be verified

1. $f(x) \Rightarrow x = g(x)$ (equivalence)
2. $\forall x \in [a, b], g(x) \in [a, b]$ (the domain contains the function)
3. $\forall x \in [a, b], |g'(x)| \leq K < 1$ (contraction)

Divergence occurs when: $\forall x \in [a, b], \min_{x \in [a, b]} |g'(x)| > 1$

Python code FIXE.py: for $f(x) = xe^x - 1$, $g(x) = e^{-x}$

```
1 1.00000000 1.71828183
2 0.36787944 -0.46853639
3 0.69220063 0.38309147
.....
15 0.56727623 0.00036739
16 0.56706790 -0.00020831
17 0.56718605 0.00011816
18 0.56711904 -0.00006701
iti = 18 x = 0.5671190400572149
```

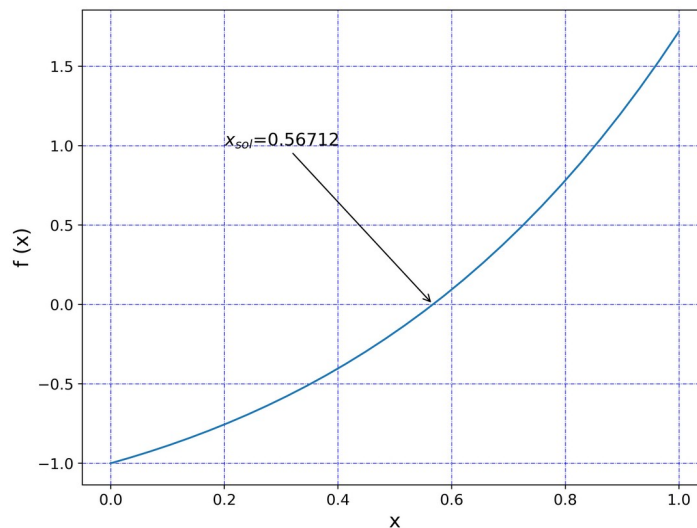


Fig.1

2 : Example of the fixed-point method

2. Solving Nonlinear Equations

2.9 Improved Iterative Approximation Method

It is possible to speed up the convergence of the successive approximation method by using the following technique:

Starting with $f(x) = 0$

We go from $x = g(x)$ to $x + \lambda x = g(x) + \lambda x$ (or $\lambda \neq -1$)

Let's also consider $x = \frac{g(x) + \lambda x}{(1 + \lambda)} = G(x)$

we choose λ such that $G'(x_n) = 0$ for each iteration, which leads us to:

$$G'(x) = \frac{g'(x) + \lambda}{(1 + \lambda)} = 0$$

Where $\lambda = -g'(x_n)$, so:

$$x_{n+1} = \frac{g(x_n) - g'(x_n)x_n}{1 - g'(x_n)} \quad (2.24)$$

In fact, this approach is an application of Newton's method in the form

$$f(x) = x - g(x) \text{ or } x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad x_{n+1} = x_n - \frac{x_n - g(x_n)}{1 - g'(x_n)}, \text{ so } x_{n+1} = \frac{g(x_n) - g'(x_n)x_n}{1 - g'(x_n)}$$

Example: Solve the equation $xe^x - 1 = 0$ using the standard successive approximation method, then refine the solution and compare (given $\epsilon = 0.001$)

Solution: Standard method of successive approximation:

$$x_0 = 1.0$$

$$x_{n+1} = e^{-x_n}$$

2. Solving Nonlinear Equations

n	x_n	$g(x_n)$	$f(x_n)$
0	1.0000000	0.3678794	1.7182818
1	0.3678794	0.6922006	-0.4685364
2	0.6922006	0.5004735	0.3830915
3	0.5004735	0.6062435	-0.1744679
4	0.6062435	0.5453958	0.1115662
5	0.5453958	0.5796123	-0.0590335
6	0.5796123	0.5601155	0.0348087
7	0.5601155	0.5711431	-0.0193080
8	0.5711431	0.5648793	0.0110887
9	0.5648793	0.5684287	-0.0062442
10	0.5684287	0.5664147	0.0035557

Table7 : Solution using the method of successive approximations.

Table8 : Solution using the method of successive approximations.

$$x_0 = 1.0, \quad x_{n+1} = G(x_n),$$

$$G(x) = \frac{e^{-x} + xe^{-x}}{1.0 + e^{-x}}$$

n	x_n	$G(x_n)$	$f(x_n)$
1	1.0000000	0.5378828	1.7182818
2	0.5378828	0.5669870	-0.0789414
3	0.5669870	0.5671433	-0.0004318
4	0.5671433	0.5671433	0.0000000
5	0.5671433	0.5671433	0.0000000

Table9 : Solution using the improved method of successive approximations

ises and Problems

2,10 Application Exercises

01: Existence and Uniqueness of Solutions Prove that the following equations have roots in the specified intervals and investigate the uniqueness of the solutions:

a) $f(x) = xe^x - 2x$ in $[0,1]$

b) $f(x) = \ln(x) - \arctg(x)$ in $[3,4]$

c) $f(x) = x + xe^x - e^x$ in $[0,1]$

02: Binary Search Method: For an error of order 10^{-6} , determine the roots of the previous equations using the binary search method

2. Solving Nonlinear Equations

03: Regula-Falsi and Secant Methods: For an error of order 10^{-6} , determine the roots of the following equations using the Regula-Falsi and Secant methods:

a) $f(x) = e^x - x - 2$ in $[1, 2]$

b) $f(x) = x - \frac{1}{2} \cos(x)$ in $\left[0, \frac{\pi}{2}\right]$

c) $f(x) = \ln(x) - x^2 - 2$ in $[1, 2]$

03: Newton-Raphson method: For an error of order 10^{-6} , determine the roots of the following equations using Newton's method:

a) $f(x) = \cos(x) - x$ in $[0, 1]$

b) $f(x) = 2 \cosh\left(\frac{x}{4}\right) - x$ in $[2, 4]$

c) $f(x) = \tan(\pi x) - x - 6$ in $[0.4, 0.48]$

04: Fixed-Point Method: For an error of order 10^{-6} , determine the roots of the preceding equations using the fixed-point method.

Problem 01 (2019–2020 Entrance Exam for Graduate Programs)

Consider the equation $f(x) = 0$ 1

or: $f(x) = x + xe^x - e^x$

1_ Show that (1) has a unique solution in $[0, 1]$

2_ Let $g(x) = \frac{e^x}{1+e^x}$

a) Show that equation (1) is equivalent to $g(x) = x$

b) the following iterative process:

$$\begin{cases} x_0 \in [0, 1] \\ x_{n+1} = g(x_n) \quad \forall n \geq 0 \end{cases}$$

Show that this process converges to the solution α and state what it corresponds to.

c) Let $x_0 = 0$. Calculate x_1 and x_2 , and find the minimum number of iterations N required to approximate α to 8.10^{-5}

3_ Is Newton's method applicable in this case?

Problem 02

Let $f(x) = xe^x - 2x$, $x \in \mathbb{R}$

1) Show that the function f has a unique minimum at α in the interval $[0, 1]$.

2) Show that Newton's algorithm can be used to find this minimum. Write the Newton sequence.

3) Calculate x_4 for $x_0 = 1.0$

4) We propose using the following scheme to determine α by the fixed-point method

$$x_{n+1} = \ln 2 - \ln(1 + x_n)$$

Show that this procedure converges.

2. Solving Nonlinear Equations

Problem 03

Let f be the function defined on $\mathbb{R}$:

$$f(x) = x - 1 + \sin(2x)$$

- 1) Show that this function has a root in the interval $[0,1]$
- 2) Investigate the convergence of the fixed-point method for $g(x) = 1 - \sin(2x)$
- 3) Does Newton's method converge? What is its order of convergence?
- 4) Give Newton's algorithm and calculate $x^{(4)}$.