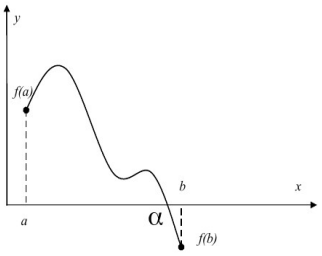
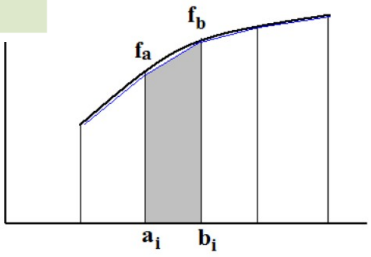
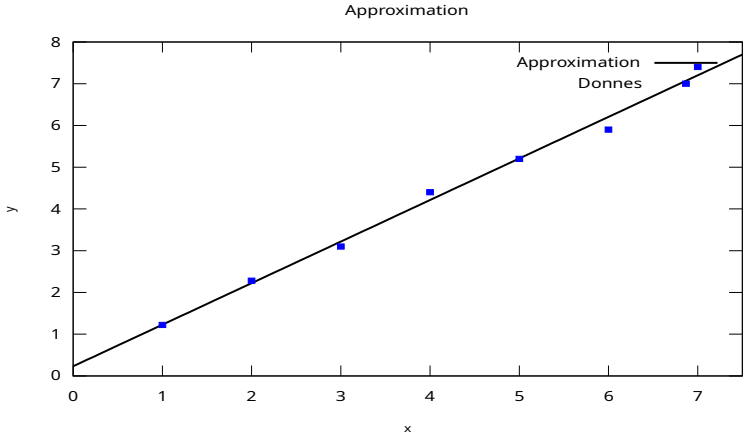
$$L_i(x)=\frac{(-1)^n}{(n-i)!\,i!\,h^n}\prod_{j=0,j\neq i}^{n}(x-x_j)$$

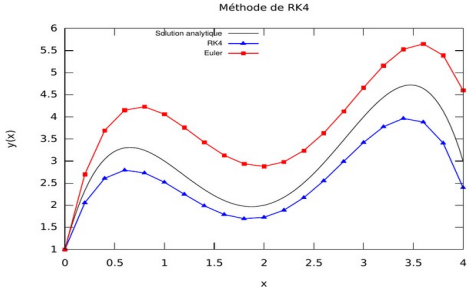


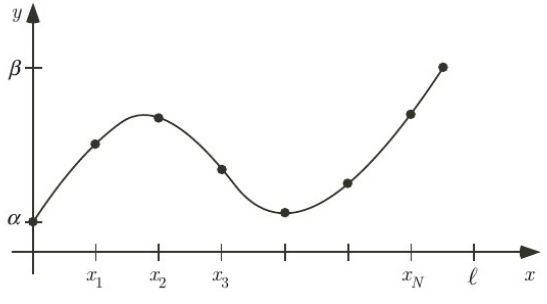




Numerical Analysis
Dr Righi Haroun

(HNSREESD 2024)

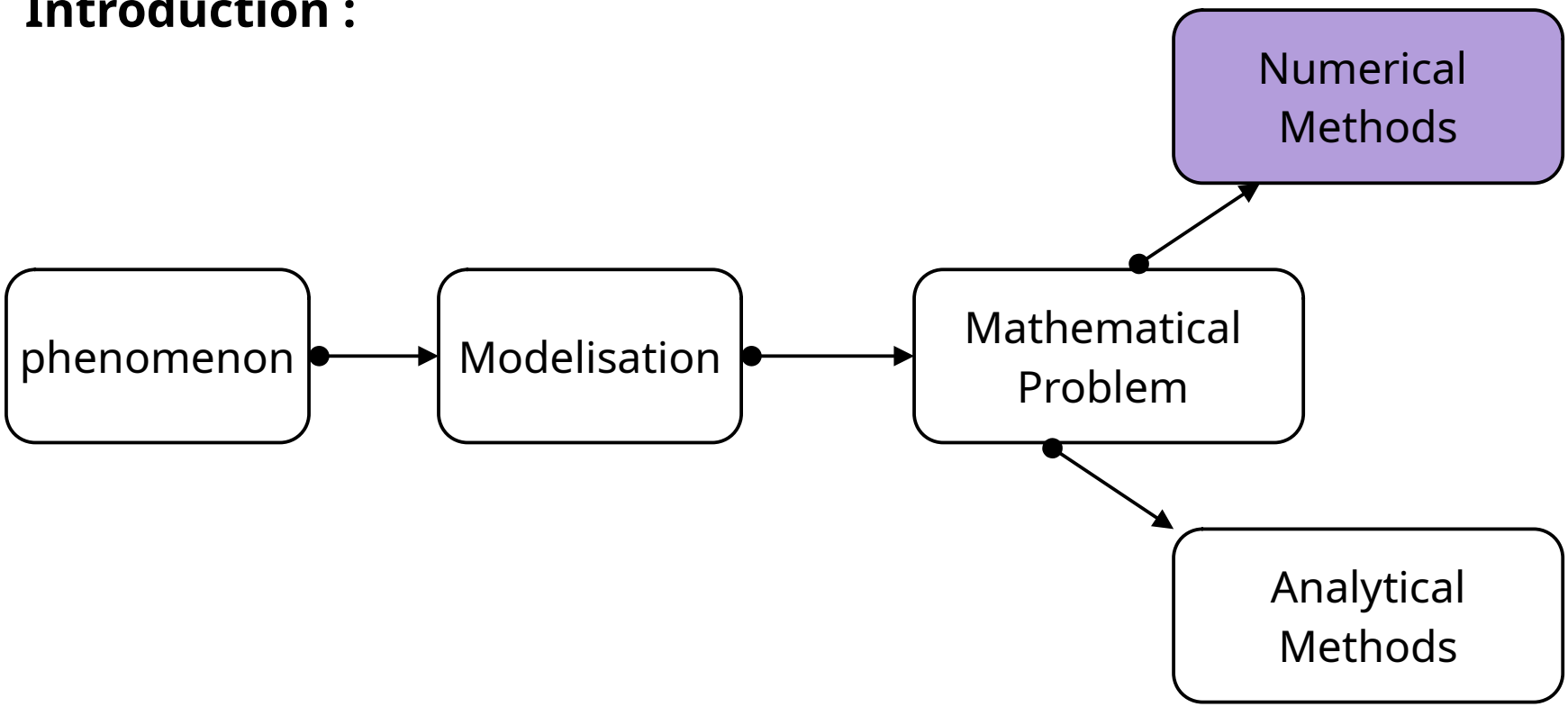




$$x_i^{k+1}=\frac{1}{a_{ii}}\left(b_i-\sum_{j=1}^{i-1}a_{ij}x_j^k-\sum_{j=i+1}^{n}a_{ii}x_j^k\right)$$

Introduction

Introduction :

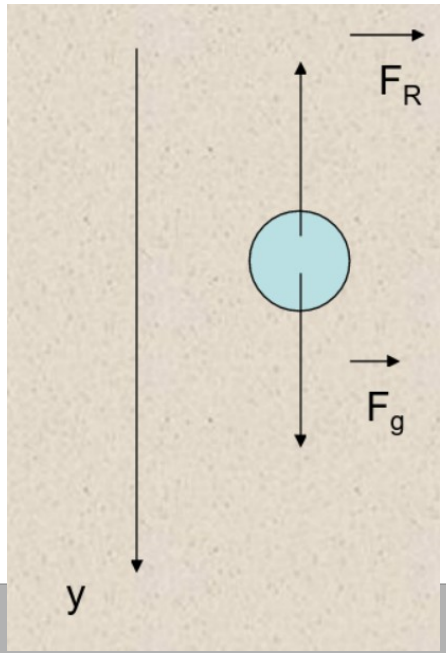


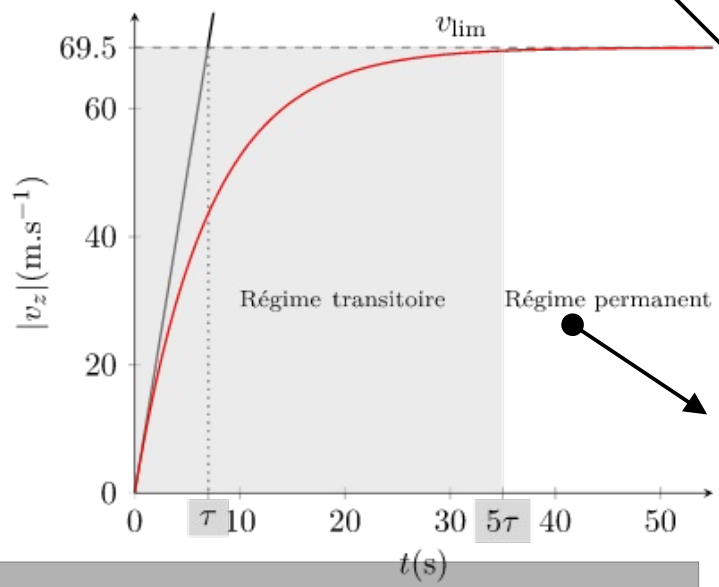
Introduction

Introduction :

free fall → Mecanics → ODE → Analytical Methods

ODE → Numerical Methods

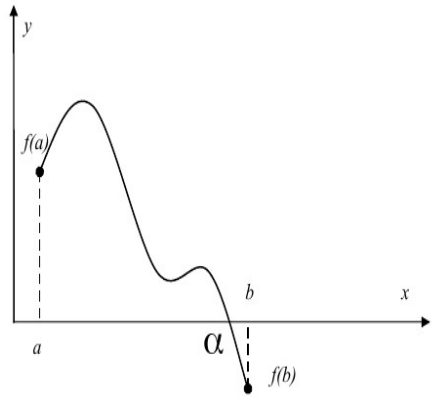




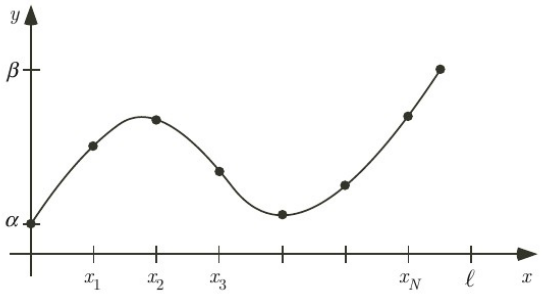
Content of the material

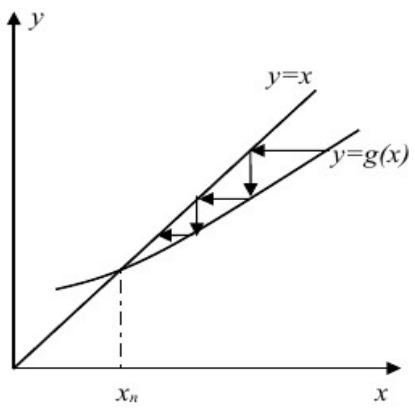
Numerical Analysis 1

Chapter I Introduction to numerical analysis, Concept of errors
Chapter II Nonlinear Equation Solving
Chapter III. Solving systems of linear equations.
_Direct methods:
_Iterative methods:

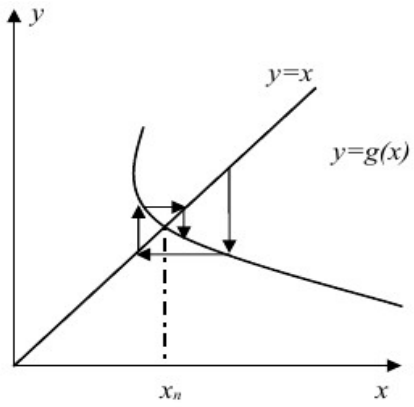


ROOT-ZERO-FIDING
A Transcendental Equation





(a)



(b)

Introduction

Introduction : Problematic

Let f : I =[a, b]⊆ R → R, find α ∈ R $f(\alpha)=0$

Definition (A Transcendental Equation) :

For a polynomial P(x), any real or complex argument x_0 such that $\mathbf{P(x_0) = 0}$ is called a root of the polynomial. For a general real-valued function **f(x)**, t is customary to use other terminology. An argument x_0 such that $f(x_0) = 0$ is called a ***zero*** of the function.

Example : We want to determine the volume V occupied by a gas whose temperature is T and whose pressure is p. The equation of state (i.e. the equation linking p, V and T) is given by

$$[P+a(n/V)^2](V-Nb)=kNT$$

Introduction

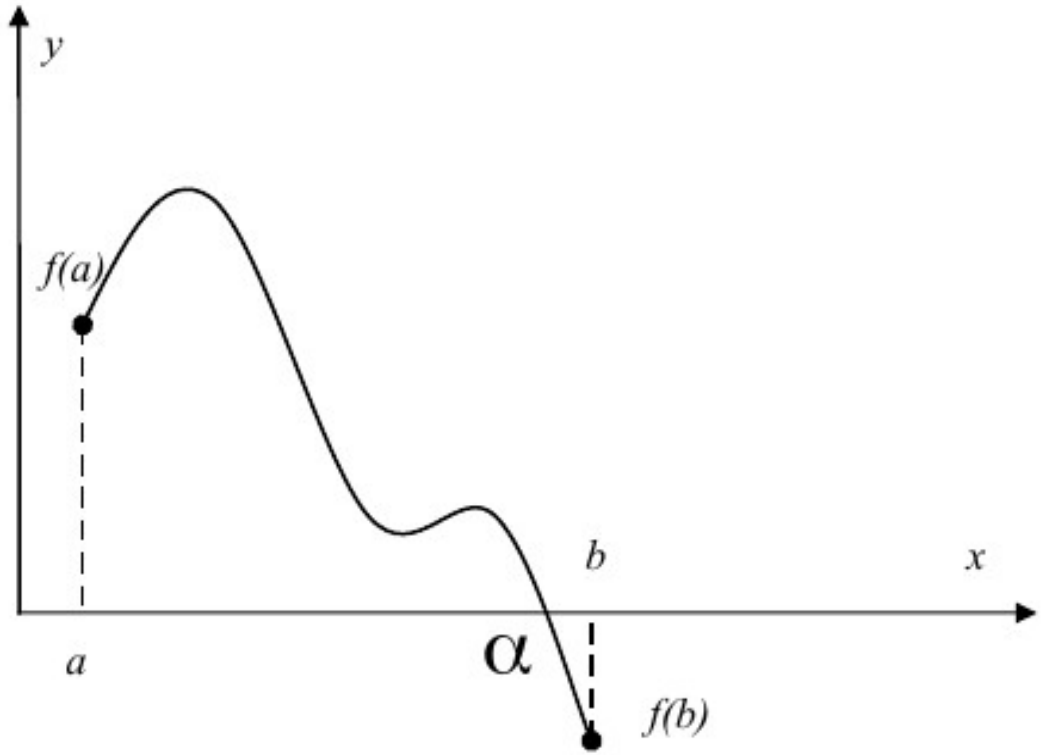
Existence of a solution **:** The approximate search for the zero of an equation is generally done in two steps

1 Solution separation: which consists of establishing segments [a, b] as tight as possible containing one and only one root.

2 Improving the accuracy or fine-tuning the approximate roots, i.e. obtaining their imposed accuracy..

Théorème I : if a continuous function f(x) takes at the extremities of the segments [a, b] opposite sign values, f(a)*f(b)<0, this segment contains at least one zero of the equation f(x)=0.

Introduction

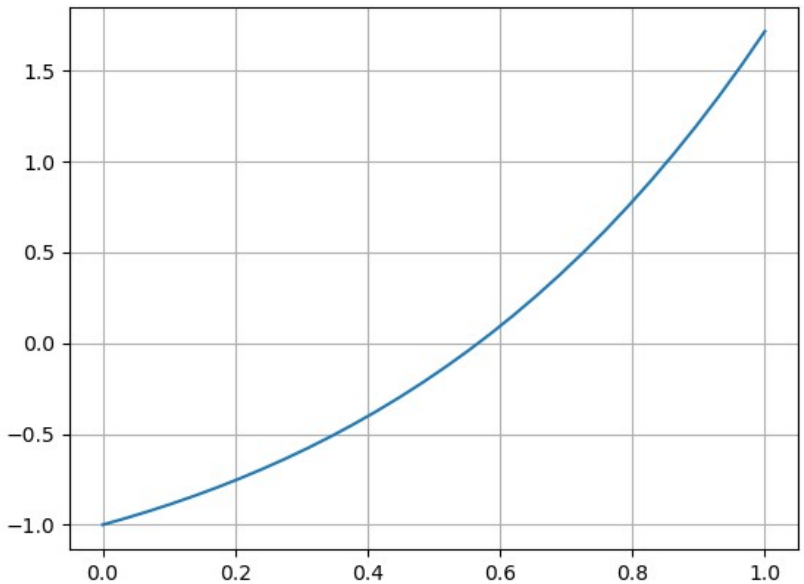


Introduction

Uniqueness: The root is unique only if the function is monotonic on the interval [a,b]

Example: Investigate the existence and uniqueness of real roots of the following function in [0,1]

$$f(x) = xe^x - 1$$



Bisection Method : (interval halving method)

Description of the method:

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

$$x_1=(a+b)/2$$

if $f(x_1)=0$ then the solution is x_1 else , we seek the solution in the other two intervals by applying Theorem I :

if $f(a).f(x_1)<0$ the needed solution is in this interval $[a, x_1]$ Denote $[a_1, b_1]$ wher $a_1 =a$ et $b_1=x_1$.

else the solution is in the other interval [x , b] named $[a_1, b_1]$ wher $a_1 = x_1$ and $b_1=b$ and we continue the search process until we obtain an interval $[a_n, b_n]$.

$$|b_n - a_n| < \epsilon$$

$$x_n=(a_n+b_n)/2$$

Bissection :

Example: Determine the root of f(x) by the method of bisection for a 4 digit :

$$f(x) = xe^{x} - 1 \qquad x \in [0,1]$$

Solution :

Bissection :

I	a	b	x	f(x)
0	0.000000000000	1.000000000000	0.500000000000	-0.175639364650
1	0.500000000000	1.000000000000	0.750000000000	0.587750012460
2	0.500000000000	0.750000000000	0.625000000000	0.167653723395
3	0.500000000000	0.625000000000	0.562500000000	-0.012781755460
4	0.562500000000	0.625000000000	0.593750000000	0.075142355321
5	0.562500000000	0.593750000000	0.578125000000	0.030619243927
6	0.562500000000	0.578125000000	0.570312500000	0.008779997268
7	0.562500000000	0.570312500000	0.566406250000	-0.002035377862
8	0.566406250000	0.570312500000	0.568359375000	0.003363661588
9	0.566406250000	0.568359375000	0.567382812500	0.000661982766
10	0.566406250000	0.567382812500	0.566894531250	-0.000687236956
11	0.566894531250	0.567382812500	0.567138671875	-0.000012761992
12	0.567138671875	0.567382812500	0.567260742188	0.000324576657
13	0.567138671875	0.567260742188	0.567199707031	0.000155898900

Iterations Numbers:

If the required precision is: ε then the number of iterations in the bisection method is

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

Algorithme :

```
Begin
Lire a, b,
Do while  |a-b|>ε
x=(a+b)/2
If  f(x)=0 proces stop
End if
If  f(a).f(x)<0        b=x
Else                   a=x
End if
End do
```

Python Code :

```
from math import *
def f(x): return x*exp(x)-1.0
A=0.0
B=1.0
Tol=1.0E-4
i=0
while abs (b-a)>tol:
  x1 = (a+b)/2.0
  if f(a)*f(x1)<0:
    b=x1
  else:
    a=x1
  print (f"{i:2d}"," ",f"{a:2.8f}"," ",f"{b:2.8f}"," ",f"{x1:2.8f}", " ", f"{f(x1):2.8f}")
  i=i+1
print ("num iti=",1)
print ("x=",x1)
```

Python Code :

```
from math import *
def f(x): return x*exp(x)-1
a=0
b=1
tol=1E-4
k=0
for i in range(30):
  x1 = (a+b)/2.0
  if f(a)*f(x1)<0:
    b=x1
  else:
    a=x1
  print (i," ",a," ",b," ",x1, " ", f(x1))
  if abs(b-a) < tol:
    k=i
    break
  else:
    continue
print ("num iti=",k)
print ("x=",x1)
```

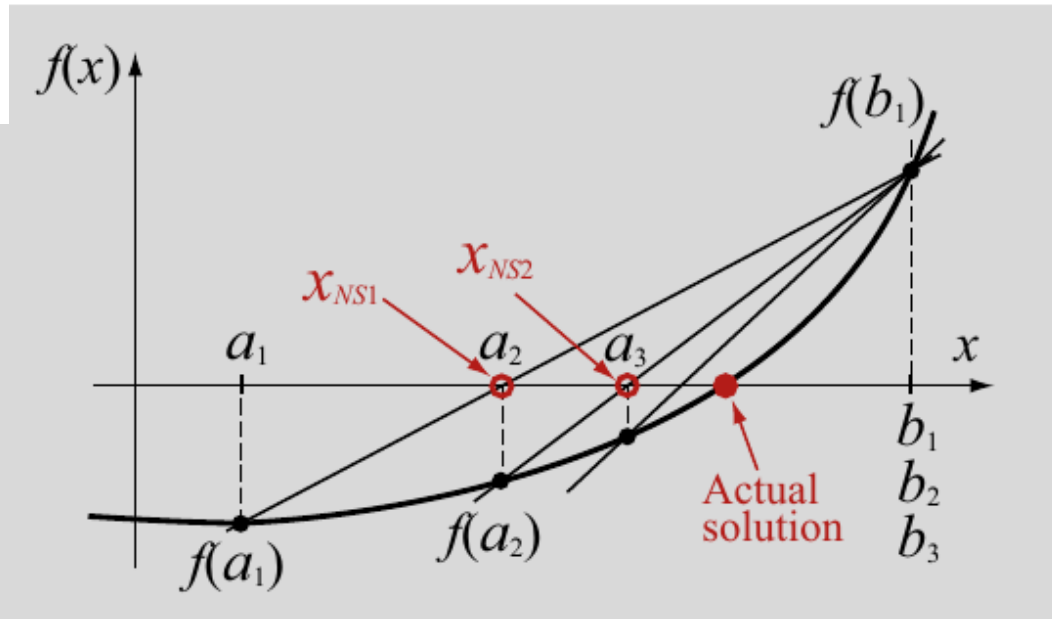
Regula Falsi Methode

In this method the search for the solution of f(x)=0 in the interval [a, b] goes through 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 into the ratio –f(a)/f(b) we thus obtain the value approximated to the root:

$$y = \frac{f(b) - f(a)}{b - a}(x - b) + f(b)$$

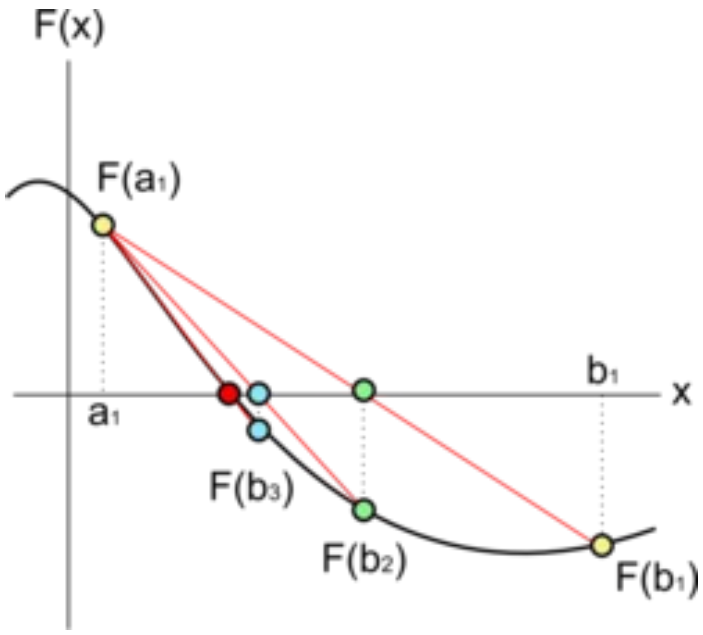
$$x = a - f(a)\frac{(b-a)}{f(b)-f(a)}$$

$$x_{NS} = \frac{af(b) - bf(a)}{f(b) - f(a)}$$



Regula Falsi Method

If the required precision is: ε then the number of iterations in the bisection method is



Begin
read *a, b,*
do
x=(af(b)-bf(a))/(f(b)-f(a))
***if f(x)=0 stop* processing**
endif
if *f(a).f(x)<0* *b=x*
else *a=x*
endif
while |f(x)|>e

Regula Falsi Methode

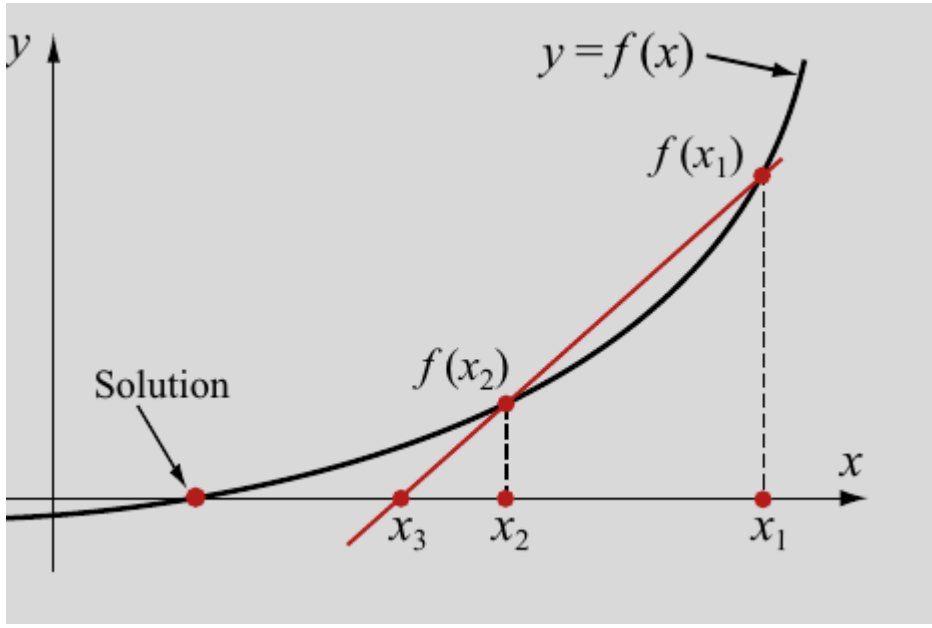
Order of Convergence: A root-finding method providing a sequence $\{x_k\}$ with $\lim_{k\to\infty} x_k = \alpha$ has order of convergence $p \geq 1$ if

$$\lim_{k\to\infty} \frac{|x_{k+1} - \alpha|}{|x_k - \alpha|^p} = C, \tag{1.11}$$

for some positive *bounded* constant C, usually termed as the *asymptotic error constant*. In particular, if $p = 1$, then C must be *smaller than unity* for the criterion to be applicable.

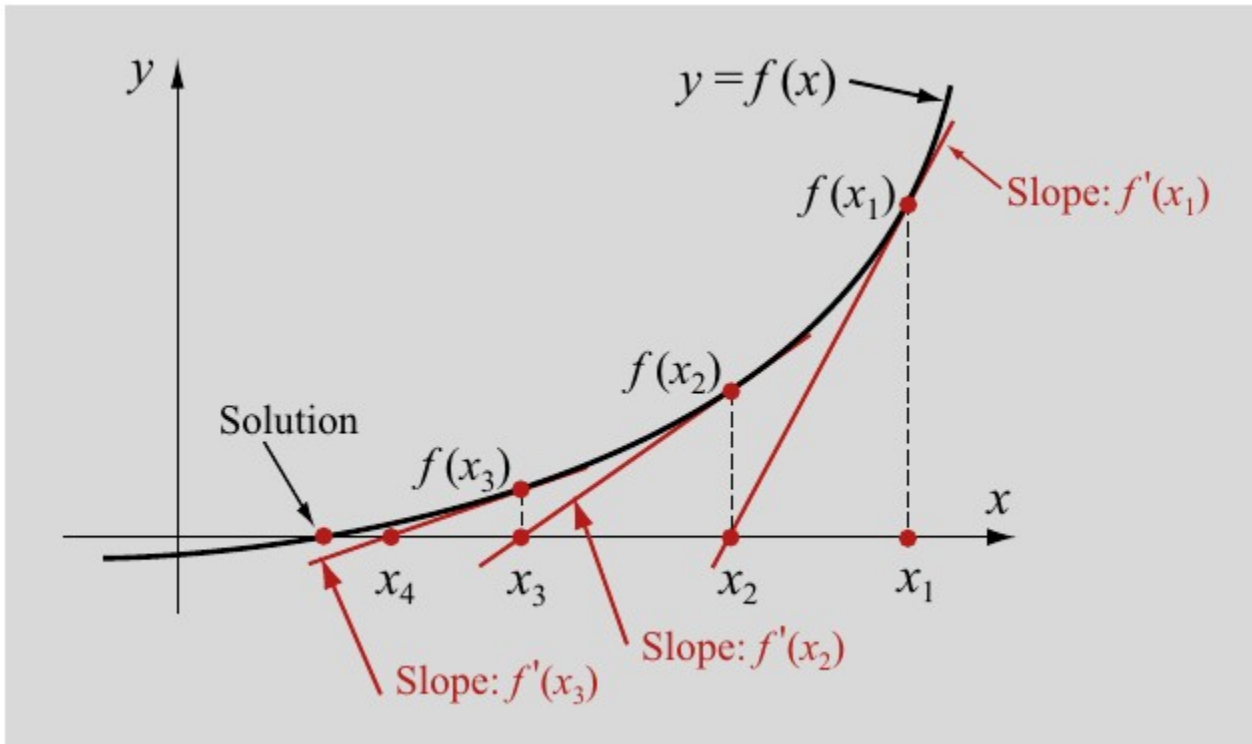
Secant's Method

$$x_{i+1} = x_i - \frac{f(x_i)(x_{i-1} - x_i)}{f(x_{i-1}) - f(x_i)}$$

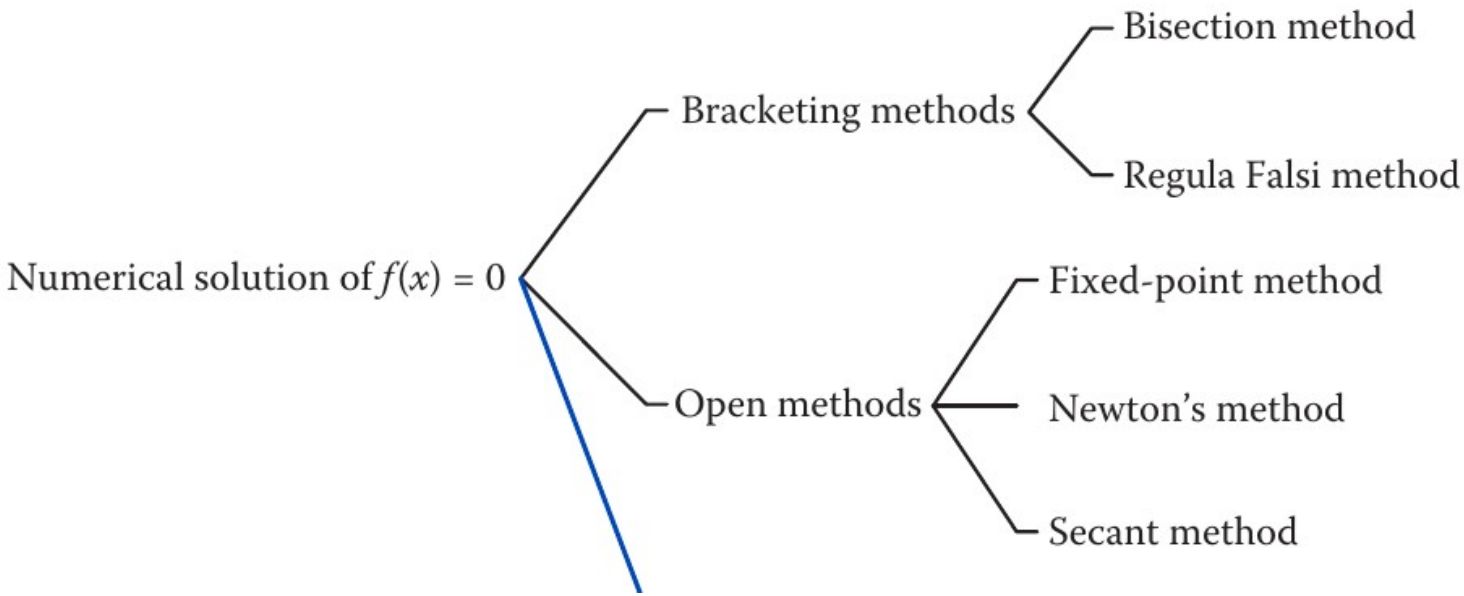


Newton-Raphson's Method

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$



Newton-Raphson's Method



Numerical CODE

Newton-Raphson's Method

$$f(x)=f(x_0)+(x-x_0)f'(x_0)+\frac{(x-x_0)^2}{2!}f''(x_0)+\frac{(x-x_0)^3}{3!}f^{[3]}(x_0).....$$

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

$$f(x)=f(x_0)+(x-x_0)f'(x_0) \qquad f(x)=0$$

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

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

Newton-Raphson's Method

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

$$x_0 = given$$

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Termination Criteria and errors estimation

$$|x_{k+1} - x_k| < \epsilon$$

$$\left|\frac{x_{k+1} - x_k}{x_k}\right| < \epsilon$$

$$|f(x_k)| < \epsilon$$

Newton-Raphson's Method

Algorithm

Begin
Def f,f'
read x0,
k=0
do
$x_{k+1}=x_k-f(x_k)/f'(x_k)$
k=k+1
while $|(x_k-x_{k-1})|>e$

Newton-Raphson's Method

Python Code:

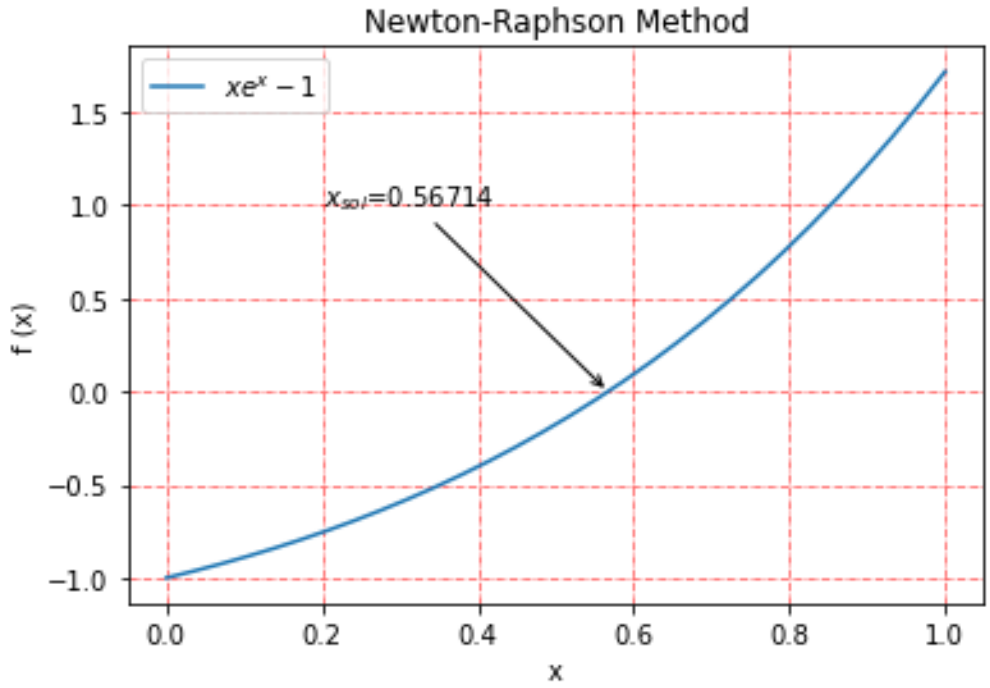

```
from math import *
import numpy as np
def f(x): return x*np.exp(x)-1
def df(x): return (1.0+x)*np.exp(x)
x = np.zeros(100, float)
tol=1E-4
x[0]=0.0
k=0
for i in range(1,30,1):
  x[i]=x[i-1]-f(x[i-1]) /df(x[i-1])
  print (f"{i:2d}"," ",f"{x[i]:2.8f}"," ",f"{f(x[i]):2.8f}")
  if np.abs(x[i]-x[i-1]) < tol:
    k=i
    break
print ("iti=",k,"x=",x[k])
```

Newton-Raphson's Method

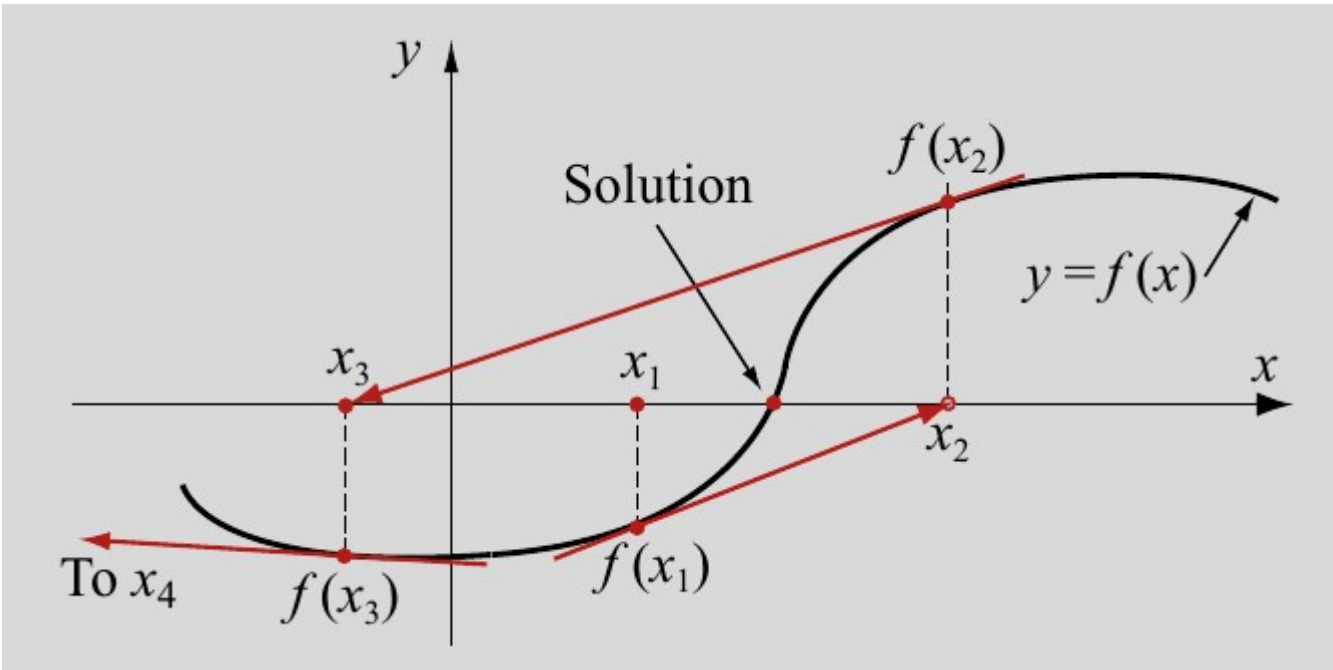
Python Code:

I	x(i)	f(x(i))
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

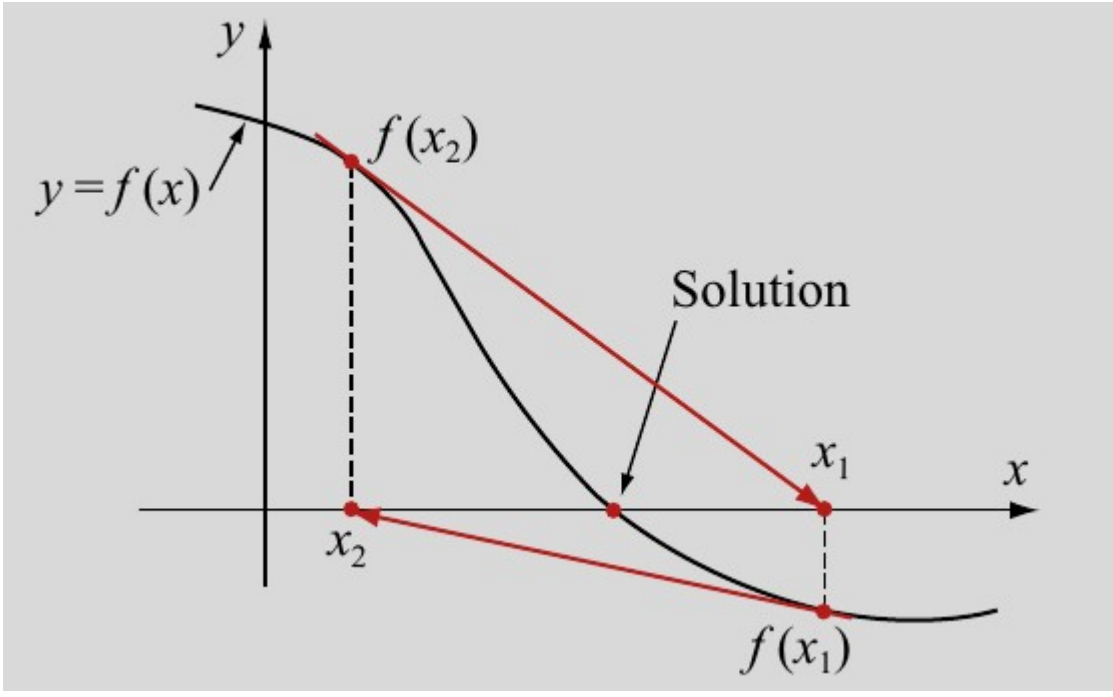
iti= 5 x= 0.567143296530295 9



Newton-Raphson's Method



Newton-Raphson's Method



Newton-Raphson's Method

Order of Convergence: A root-finding method providing a sequence $\{x_k\}$ with $\lim_{k\to\infty} x_k = \alpha$ has order of convergence $p \geq 1$ if

$$\lim_{k\to\infty} \frac{|x_{k+1} - \alpha|}{|x_k - \alpha|^p} = C, \tag{1.11}$$

for some positive *bounded* constant C, usually termed as the *asymptotic error constant*. In particular, if $p = 1$, then C must be *smaller than unity* for the criterion to be applicable.

Newton-Raphson's Method

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

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

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

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

$$C=\left|\frac{f''(\eta)}{2f'(x_n)}\right|$$

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

Newton-Raphson's Method

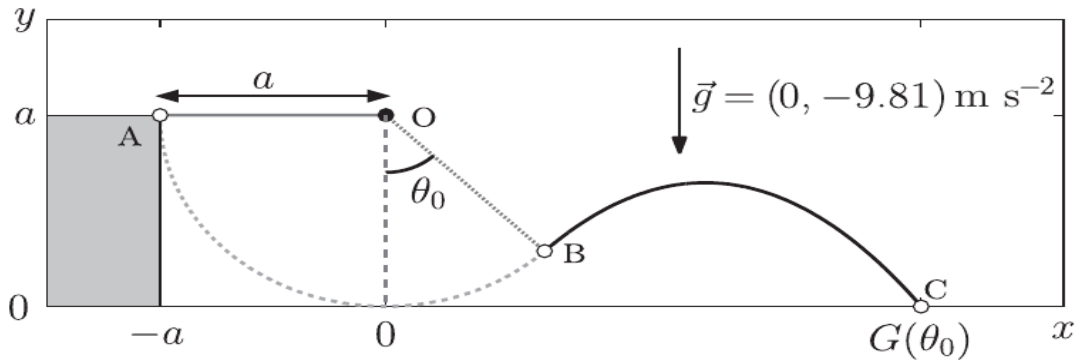
Newton-Raphson's Method

3.5.3 Modified Newton's Method for Roots with Multiplicity 2 or Higher

If r is a root of $f(x)$ and r has a multiplicity 2 or higher, then convergence of the sequence generated by Newton's method is linear; see Theorem 3.2. In these situations, Newton's method may be modified to improve the speed of convergence. The modified Newton's method designed for roots of multiplicity 2 or higher is described as

$$x_{n+1} = x_n - \frac{f(x_n)f'(x_n)}{[f'(x_n)]^2 - f(x_n)f''(x_n)}, \quad n = 1,2,3, \ \ldots, \quad x_1 = \text{initial point}$$

Newton-Raphson's Method



(a) Show that $G(\theta_0) = a\{(1 + 2\cos^2\theta_0)\sin\theta_0 + 2\sqrt{\cos^3\theta_0(1 - \cos^3\theta_0)}\}$.
(b) Find the angle θ_0 for which $G(\theta_0) = 20$ m.
(c) What is the angle θ_0 so that the acrobat maximizes the range G?
Provide your numerical answers with at least four exact figures. Advice: for parts (b) and (c), consider the alternative equation

$$F(z) = a\left\{(1 + 2z^2)\sqrt{1 - z^2} + 2\sqrt{z^3(1 - z^3)}\right\}, \quad \text{where} \quad z = \cos\theta_0.$$

Newton-Raphson's Method

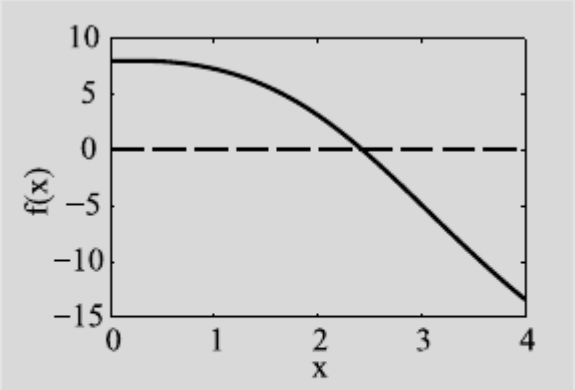


Figure 3-8: A plot of the function $f(x) = 8 - 4.5(x - \sin x)$.

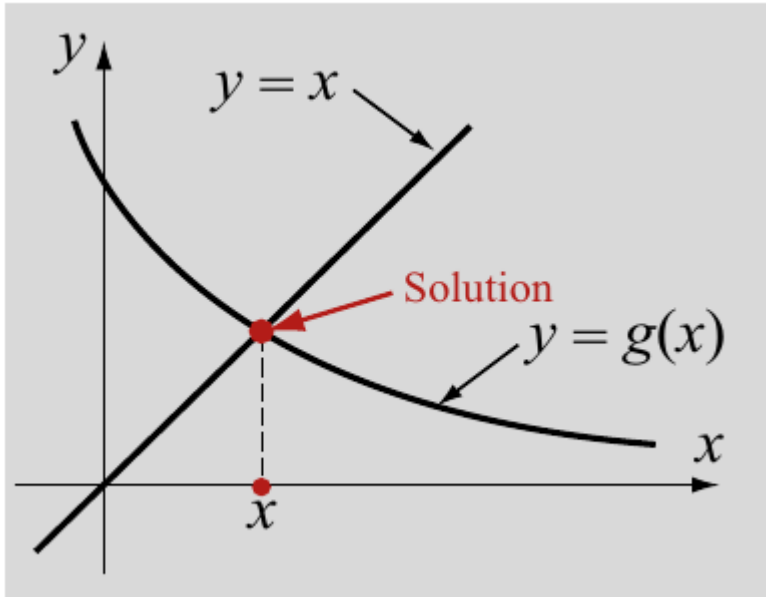
The fixed-point iteration method

Algorithm fixed-point iteration method.

$$f(x) = 0 \longrightarrow x = g(x)$$

x_0

$x_{i+1} = g(x_i)$



$$|x_{i+1} - x_i| \leq \varepsilon_1 \qquad \text{and/or} \qquad |f(x_{i+1})| \leq \varepsilon_2$$

The fixed-point iteration method

Algorithm fixed-point iteration method.

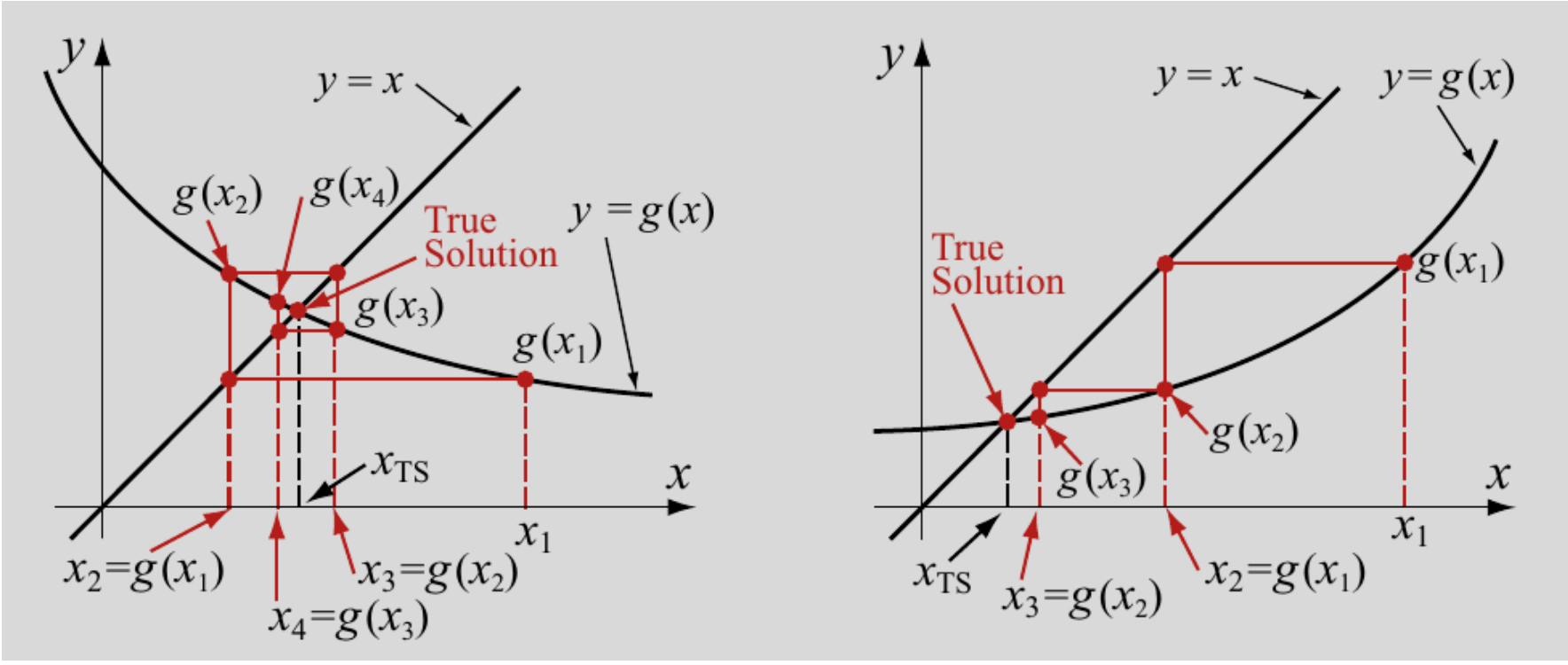
$$x_1 = \frac{1 + \frac{2}{1}}{2} = \frac{3}{2}.$$

(a)

(b)

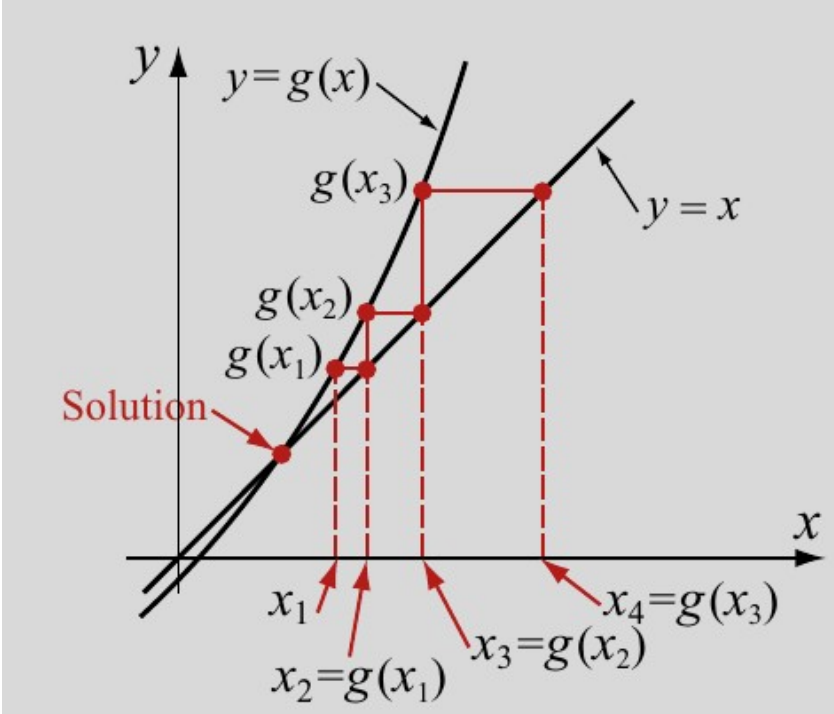
The fixed-point iteration method

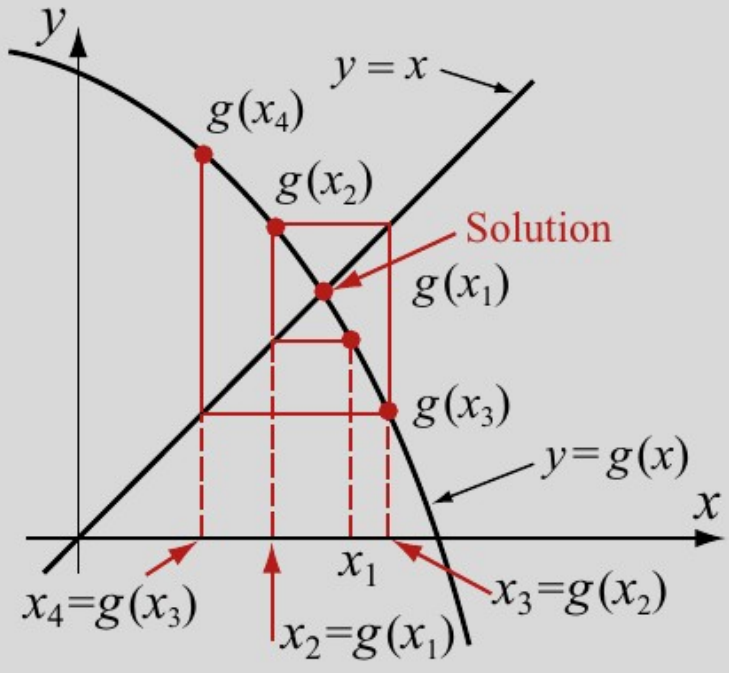
Convergence of the fixed-point iteration method.



The fixed-point iteration method

Divergence of the fixed-point iteration method.





The fixed-point iteration method

Divergence of the fixed-point iteration method.

Definition 15.10 (contraction)**.** Suppose that $-\infty < a < b < \infty$ and $g \in C([a, b])$. We say that g is **Lipschitz continuous**[1] on $[a, b]$ if and only if there exists a constant $L > 0$ such that

$$|g(x) - g(y)| \leq L|x - y|, \quad \forall x, y \in [a, b],$$

and the associated L is called the **Lipschitz constant**. The function g is called a **contraction** on $[a, b]$ if and only if it is Lipschitz on $[a, b]$ and its associated Lipschitz constant, L, satisfies $L \in (0, 1)$.

The fixed-point iteration method

Theorem 15.12 (uniqueness)**.** *Suppose that* $-\infty < a < b < \infty$, $g \in C([a, b])$, *and* $g([a, b]) \subseteq [a, b]$. *If* g *is a contraction on* $[a, b]$, *then* g *has a unique fixed point* $\xi \in [a, b]$. *Furthermore, the sequence* $\{x_k\}_{k=0}^{\infty}$ *generated by* (15.2) *converges to* ξ *for any starting value* $x_0 \in [a, b]$.

Proof. Theorem 15.9 guarantees the existence of at least one fixed point $\xi \in [a, b]$. Suppose that $\eta \in [a, b]$ is another fixed point. Since g is a contraction,

$$|\xi - \eta| = |g(\xi) - g(\eta)| \leq L|\xi - \eta|.$$

Therefore,

$$0 \leq (1 - L)|\xi - \eta| \leq 0,$$

which proves that $\xi = \eta$. Hence, the fixed point $\xi \in [a, b]$ is unique.

The fixed-point iteration method

The fixed-point iteration method converges if, in the neighborhood of the fixed point, the derivative of $g(x)$ has an absolute value that is smaller than 1 (also called Lipschitz continuous):

$$|g'(x)| < 1 \qquad (3.30)$$

The fixed-point iteration method

$$f(x) = xe^x - 1$$

$X_0 = 1.0$

$x_{i+1} = \exp(-x_i)$

I	x	g(x)
1	1.00000000	1.71828183
2	0.36787944	-0.46853639
3	0.69220063	0.38309147
4	0.50047350	-0.17446790
5	0.60624354	0.11156623
6	0.54539579	-0.05903351
.....		
14	0.56690891	-0.00064752
15	0.56727623	0.00036739
16	0.56706790	-0.00020831
17	0.56718605	0.00011816
18	0.56711904	-0.00006701