MSDM5004
Numerical Methods and Modeling in Science
Spring 2024

Lecture 1

Prof Yang Xiang
Hong Kong University of Science and Technology

Introduction

Numerical solutions

Purpose:

To understand and design numerical algorithms

Chapter 1

Computer Representation of Numbers

Reference: Numerical Computing with IEEE Floating Point Arithmetic, M. L. Overton, SIAM, 2001.

1. Decimal and binary numbers

Decimal:

$$4271.325 = 4\times 10^3 + 2\times 10^2 + 7\times 10^1 + 1\times 10^0 + 3\times 10^{-1} + 2\times 10^{-2} + 5\times 10^{-3}$$

base: 10

digit (bit): 0, 1,2, ..., β-1
where β is the base

Binary:

$$\frac{11}{2} = (101.1)_2 = 1\times 4 + 0\times 2 + 1\times 1 + 1\times \frac{1}{2}$$

base: 2

2. Floating point representation

Floating point representation is based on exponential notation

Decimal:

$$x = \pm d_1.d_2d_3\cdots d_k \times 10^n$$

$$1 \le d_1 \le 9,\ 0 \le d_i \le 9,\ i = 2, \cdots, k,\ n \text{ integer.}$$

$$4271.325 = 4.271325 \times 10^3$$

Binary:

$$x = (\pm 1.b_1b_2\cdots b_{p-1} \times 2^E)_2$$

base 2

$$b_i = 0 \text{ or } 1,\ i = 1, 2, \cdots, p-1,\ E \text{ integer}$$

$$\frac{11}{2} = (1.011)_2 \times 2^2$$

$$\boxed{\frac{11}{2} = (101.1)_2 = 1 \times 4 + 0 \times 2 + 1 \times 1 + 1 \times \frac{1}{2}}$$

3. Machine numbers

Base 2

IEEE floating point representation

Single format 32 bits

$$x = \pm(1.b_1 b_2 \ldots b_{p-2} b_{p-1})_2 \times 2^E$$

0	ebits(2)	01100000000000000000000

1 bit for the sign

8 bit for the exponent E

23 bit for the fraction

$b_1 b_2 \cdots b_{p-1}$

$$-126 \leq E \leq 127$$

from 00000001 to 11111110

The example is for $\frac{11}{2} = (1.011)_2 \times 2^2$

Double format 64 bits

$$x = \pm(1.b_1 b_2 \ldots b_{p-2} b_{p-1})_2 \times 2^E$$

$\pm$	$a_1 a_2 a_3 \ldots a_{11}$	$b_1 b_2 b_3 \ldots b_{52}$

Exponent $-1022 \le E \le 1023$

Range of machine numbers

Format	$E_{\min}$	$E_{\max}$	$N_{\min}$	$N_{\max}$
Single	-126	127	$2^{-126} \approx 1.2 \times 10^{-38}$	$\approx 2^{128} \approx 3.4 \times 10^{38}$
Double	-1022	1023	$2^{-1022} \approx 2.2 \times 10^{-308}$	$\approx 2^{1024} \approx 1.8 \times 10^{308}$

Machine numbers are discrete on the real axis

Special machine numbers

$$+0,\ -0,\ +\infty,\ -\infty,\ \mathrm{NaN}$$

not a number, e.g. 0/0

Machine epsilon

The gap between 1 and the next larger floating point number.

Format	Precision	Machine Epsilon
Single	$p = 24$	$\epsilon = 2^{-23} \approx 1.2 \times 10^{-7}$
Double	$p = 53$	$\epsilon = 2^{-52} \approx 2.2 \times 10^{-16}$

$$x = \pm(1.b_1 b_2 \ldots b_{p-2} b_{p-1})_2 \times 2^E$$

4. Rounding and significant digits

Only finite digits can be kept (p=53 in double precision) in the computer.

$$x = (1.b_1 b_2 \ldots b_{p-1} b_p b_{p+1} \ldots)_2 \times 2^E$$

Rounding to x_- or x_+ (usually round to the nearest).

$$x_- = (1.b_1 b_2 \ldots b_{p-1})_2 \times 2^E$$

$$x_+ = ((1.b_1 b_2 \ldots b_{p-1})_2 + (0.00\ldots01)_2) \times 2^E$$

i.e. $\mathrm{fl}(x) = x_-$ or x_+

floating point

Significant digits

The single precision $p = 24$ corresponds to approximately 7 significant decimal digits.

$$2^{-24} \approx 10^{-7}.$$

$$\pi = 3.141592|653\ldots$$

The double precision $p = 53$ corresponds to approximately 16 significant decimal digits.

5. Absolute and relative errors

Suppose that p^* is an approximation to p.

The **absolute error** is $|p - p^*|$

the **relative error** is $\dfrac{|p - p^*|}{|p|}$, provided that $p \neq 0$.

6. Rounding errors

absolute error $\left| fl(y) - y \right|$

relative error $\left| \dfrac{fl(y) - y}{y} \right|$

7. Loss of significance

$$fl(x) = 0.d_1 d_2 \ldots d_p \alpha_{p+1} \alpha_{p+2} \ldots \alpha_k \times 10^n.$$ k digits

$$fl(y) = 0.d_1 d_2 \ldots d_p \beta_{p+1} \beta_{p+2} \ldots \beta_k \times 10^n.$$ k digits

$$fl(fl(x) - fl(y)) = 0.\sigma_{p+1} \sigma_{p+2} \ldots \sigma_k \times 10^{n-p}$$

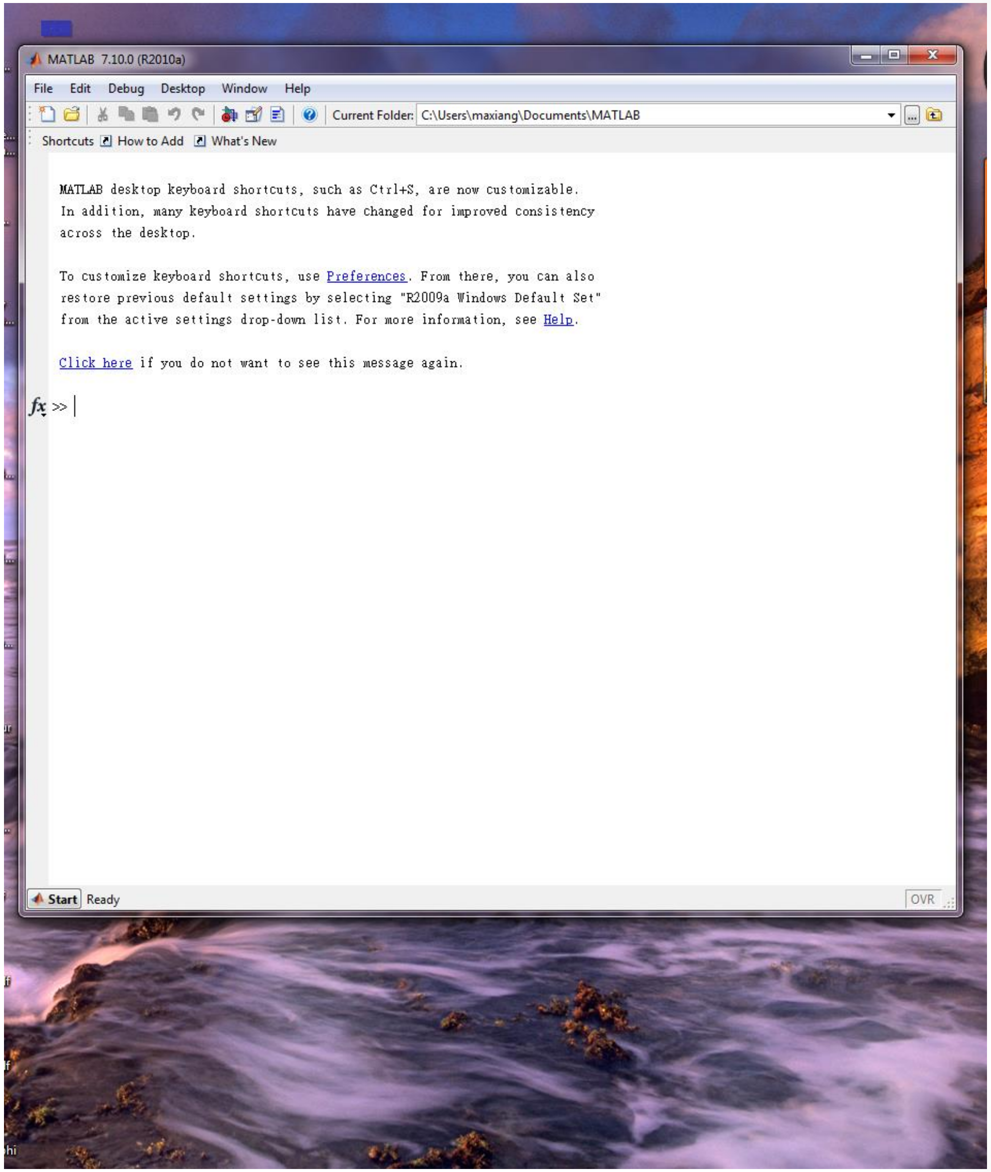
k-p digits

where

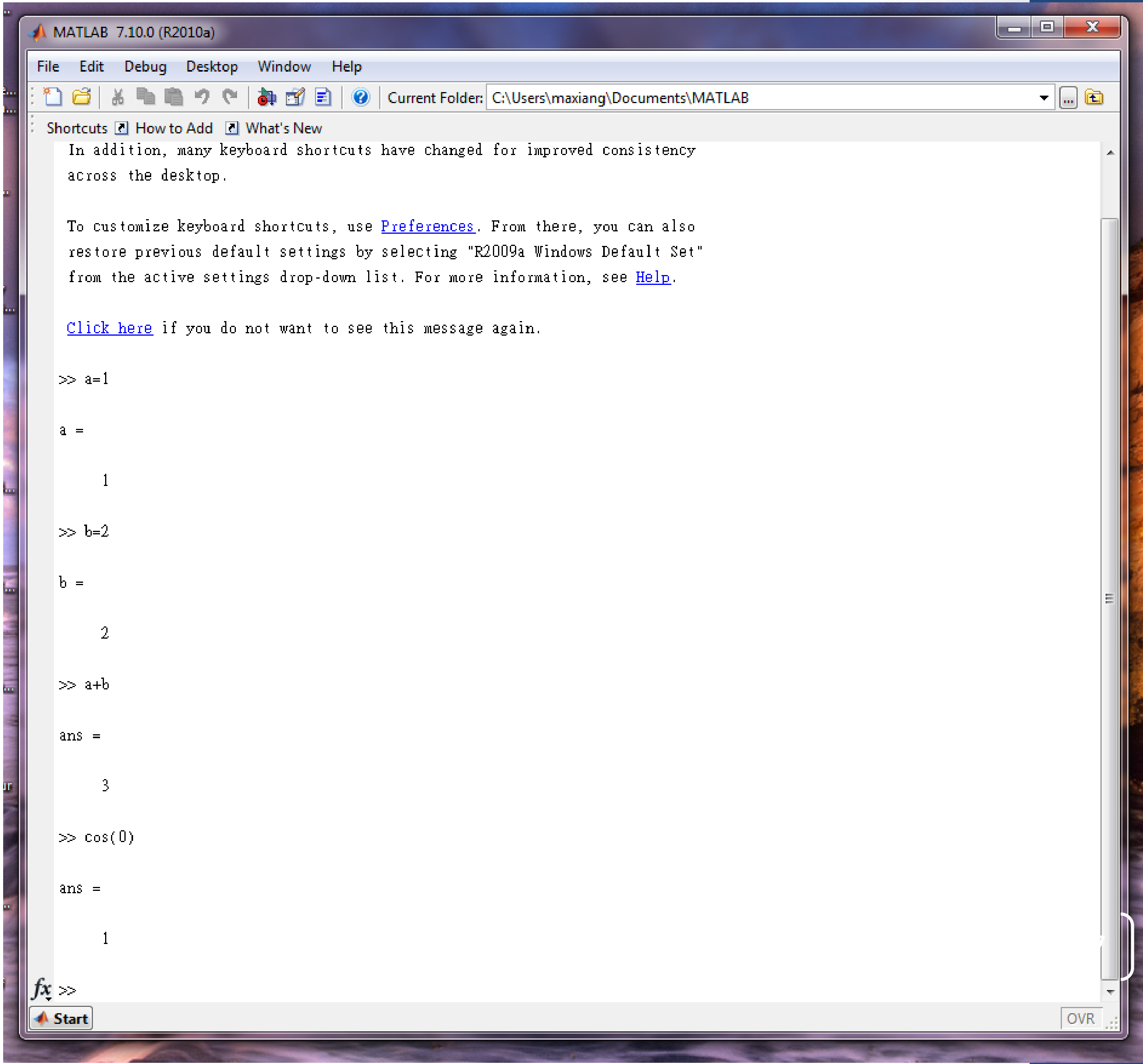
$$0.\sigma_{p+1} \sigma_{p+2} \ldots \sigma_k = 0.\alpha_{p+1} \alpha_{p+2} \ldots \alpha_k - 0.\beta_{p+1} \beta_{p+2} \ldots \beta_k$$

MATLAB Tutorial

Command window



Evaluating variables and functions



Matrices and operations

```
>> a=[1 2;3 4]

a =

     1     2
     3     4

>> a'

ans =

     1     3
     2     4

>> a*a

ans =

     7    10
    15    22

>> inv(a)

ans =

   -2.0000    1.0000
    1.5000   -0.5000

>> b=[10;1]

b =

    10
     1

>> a*b

ans =

    12
    34

>> b'*a*b

ans =

   154
```

Sovling matrix equation ax=b

$x=a^{-1}b$

```
>> a=[1 2;3 4]

a =

     1     2
     3     4

>> b=[10;1]

b =

    10
     1

>> inv(a)*b

ans =

  -19.0000
   14.5000

>>
```

Access elements in a matrix or vector

```
>> a=[1 2;3 4]
a =
    1    2
    3    4

>> b=[10;1]
b =
   10
    1

>> a(2,1)

ans =

    3

>> b(1)

ans =

   10
```

power

```
>> a^2

ans =

    7   10
   15   22

>> a^3

ans =

   37   54
   81  118
```

Matrices and operations

```
>> x=1:2:11

x =

     1     3     5     7     9    11

>> y=sin(x)

y =

    0.8415    0.1411   -0.9589    0.6570    0.4121   -1.0000

fx >>
```

Start

For-loop

e.g. Compute $$\sum_{n=1}^{20} \frac{1}{n^3}$$

```
s=0;
Nt=20;
for i=1:Nt
s=s+1/i^3;
End

>> s
s =
   1.2009
>>
```

while-loop

```
s=0;
Nt=20;
i=1;
while i<=Nt
s=s+1/i^3;
i=i+1;
end;


>> s
s =
   1.2009
```

Default display form: format short

```
>> pi

ans =

    3.1416
```

format long

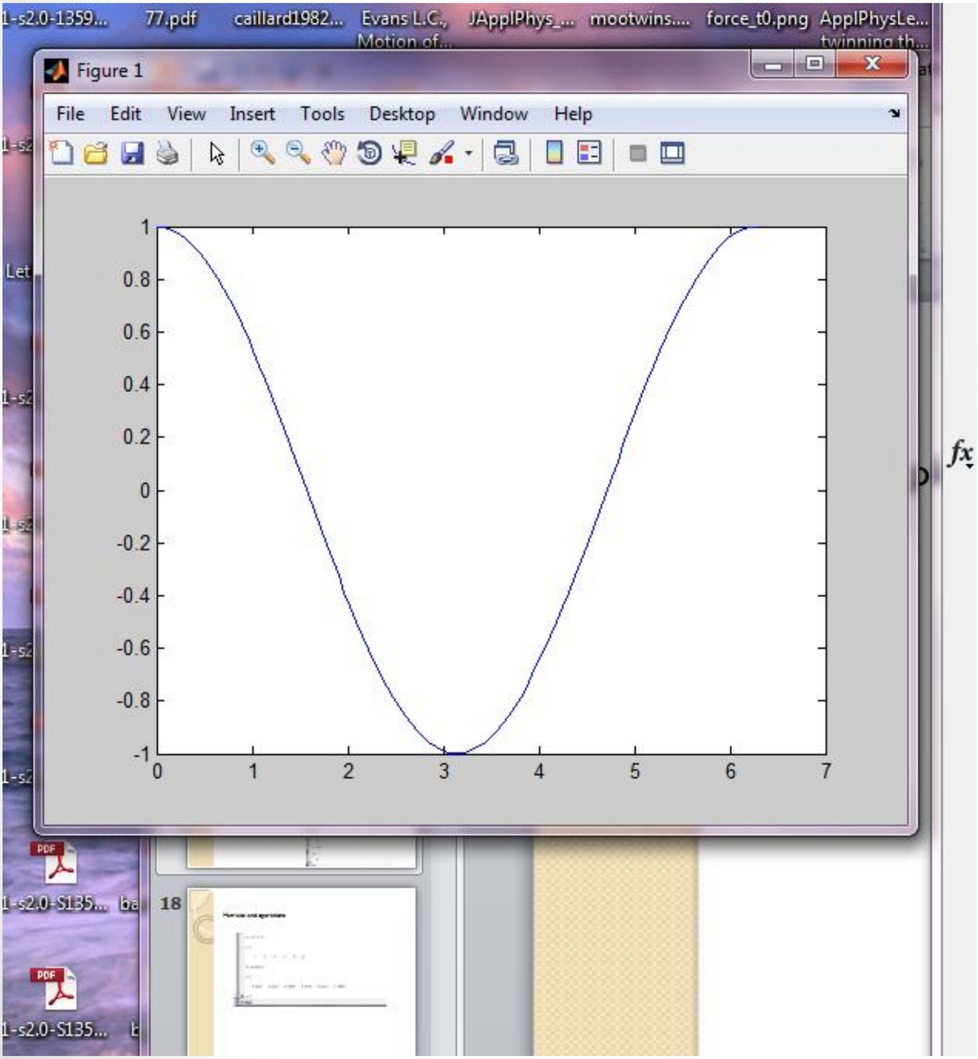
```
>> pi

ans =

   3.141592653589793
```

Remark: It is only for display. Double precision is always used in calculations.

A simple plot




```
MATLAB desktop keyboard shortcuts, such as Ctrl+S, are now customizable.
In addition, many keyboard shortcuts have changed for improved consistency
across the desktop.

To customize keyboard shortcuts, use Preferences. From there, you can also
restore previous default settings by selecting "R2009a Windows Default Set"
from the active settings drop-down list. For more information, see Help.

Click here if you do not want to see this message again.

>> x=linspace(0,2*pi,100);
>> y=cos(x);
>> plot(x,y)
>>
```

MATLAB doc

MATLAB provides a command called `doc` to show the documentation and `help` for search unknown commands. Please check out the following commands:

```
doc sum
doc sin
doc diag
doc size
doc eye
doc ones
doc linspace
doc plot
help sum
help sin
```

You are also suggested to search your questions with keyword MATLAB on the internet and try the examples you find.

Software

To use MATLAB, you need to login to Virtual Barn with VMware Horizon Client. The client can be found on the computer in Computer Barns. Alternatively, you may install the client on your own devices. When programing on Virtual Barn, remenber to connect to Academic Software as MATLAB is only installed there. Please refer to Installation Guide and User Guide for details.

ITSC webpage

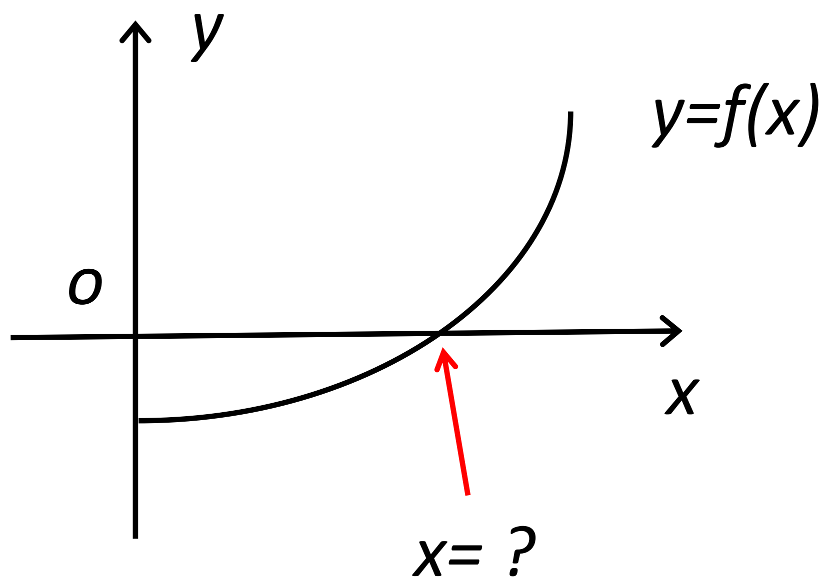
https://itsc.ust.hk/services/general-it-services/procurement-licensing/common-software-list

https://itsc.hkust.edu.hk/services/academic-teaching-support/facilities/virtual-barn

Chapter 2

Finding Roots

1. Introduction



2. General iterative algorithm

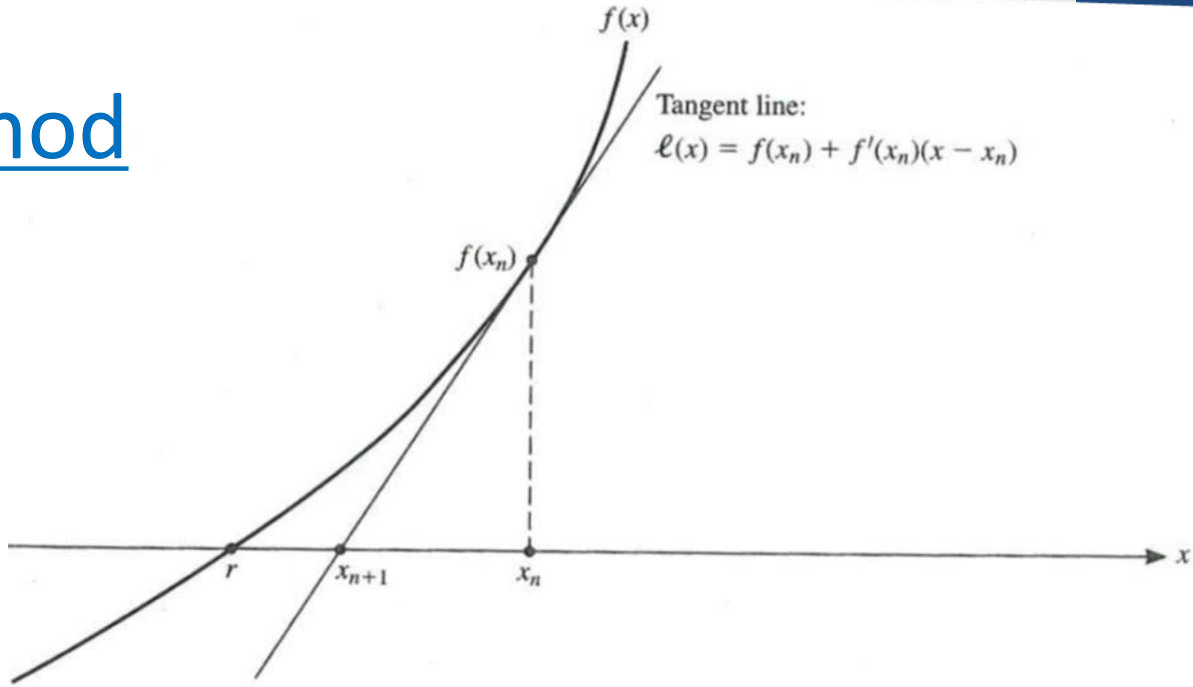
1. Specify some initial guess of the solution x_0

2. For $n=0, 1, \ldots$

   (i) If x_n is optimal, stop.

   (ii) Determine x_{n+1}, a new estimate of the solution.

3. Newton's method



From x_n to x_{n+1}

- Approximate f(x) near x_n by the tangent line *l(x)* at x_n
- Solve for *l(x)=0,* the solution is defined as x_{n+1}

Near x_n,

$$f(x) \approx l(x) = f(x_n) + f'(x_n)(x - x_n).$$

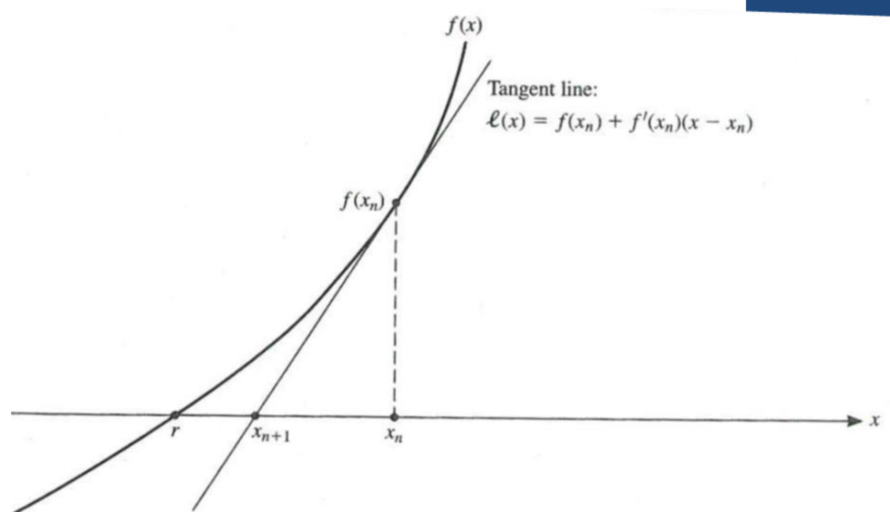
Solve for $l(x) = 0$,

$$l(x) = f(x_n) + f'(x_n)(x - x_n).$$

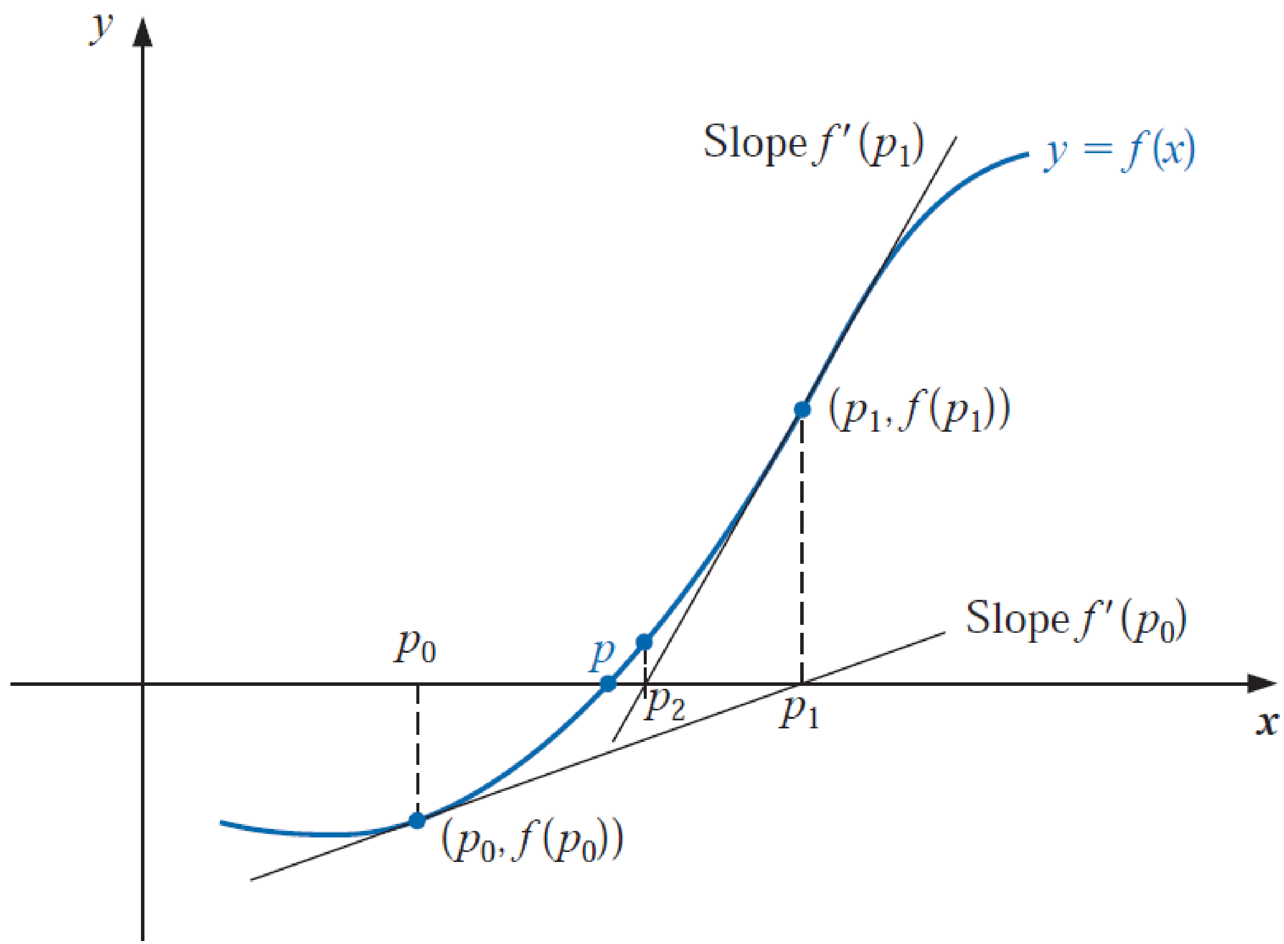
$$x = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Therefore x_{n+1} is defined as

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



The iteration starts from an initial guess x_0.



Stopping criterion

For a prespecified small $\varepsilon > 0$,

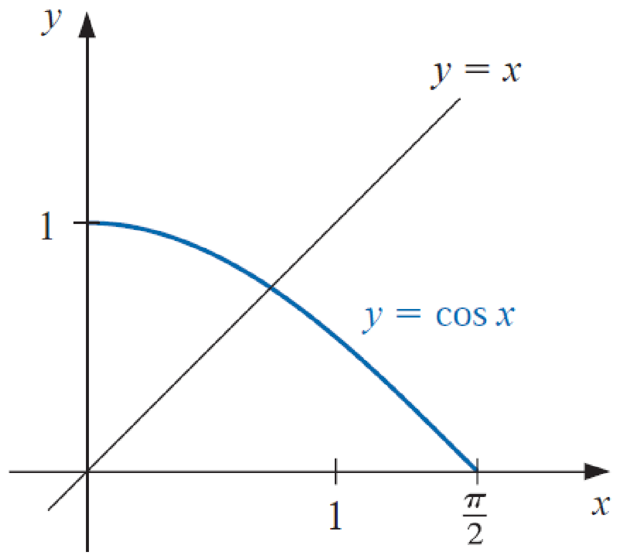
(1) $|x_{n+1} - x_n| < \varepsilon$, or

(2) $\dfrac{|x_{n+1} - x_n|}{|x_n|} < \varepsilon, \quad x_n \neq 0, \quad$ or

(3) $|f(x_{n+1})| < \varepsilon$.

An example

Solve for $f(x) = \cos x - x = 0$. The initial guess is $x_0 = \frac{\pi}{4}$. The required accuracy is $\varepsilon = 10^{-10}$.



Solution We compute

$$f'(x) = -\sin x - 1.$$

The Newton's method is

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

$n = 0,$

$$x_1 = x_0 - \frac{\cos x_0 - x_0}{-\sin x_0 - 1} = \frac{\pi}{4} - \frac{\cos\frac{\pi}{4} - \frac{\pi}{4}}{-\sin\frac{\pi}{4} - 1} = 0.7395361337.$$

$n = 1,$

$$x_2 = x_1 - \frac{\cos x_1 - x_1}{-\sin x_1 - 1} = 0.7390851781.$$

$n = 2,$

$$x_3 = x_2 - \frac{\cos x_2 - x_2}{-\sin x_2 - 1} = 0.7390851332.$$

$n = 3,$

$$x_4 = x_3 - \frac{\cos x_3 - x_3}{-\sin x_3 - 1} = 0.7390851332.$$

$$|x_4 - x_3| < 10^{-10}.$$

The solution of $f(x) = 0$ is approximately $x_4 = 0.7390851332$.

Convergence of the Newton's method

Let x_* be the solution of $f(x) = 0$.

Assume that $f \in C^2[a,b]$, and $f'(x_*) \neq 0$.

By Taylor expansion at x_n, we have

$$0 = f(x_*) = f(x_n) + f'(x_n)(x_* - x_n) + \frac{1}{2} f''(\xi)(x_* - x_n)^2, \quad (1)$$

where ξ is between x_* and x_n.

Denote the error $e_n = x_n - x_*$.

By Newton's method, we have

$$e_{n+1} = x_{n+1} - x_* = x_n - \frac{f(x_n)}{f'(x_n)} - x_* = e_n - \frac{f(x_n)}{f'(x_n)}. \quad (2)$$

Using Eq. (1), we have

$$f(x_n) = -f'(x_n)(x_* - x_n) - \frac{1}{2} f''(\xi)(x_* - x_n)^2.$$

$$\frac{f(x_n)}{f'(x_n)} = -(x_* - x_n) - \frac{f''(\xi)}{2f'(x_n)}(x_* - x_n)^2 = e_n - \frac{f''(\xi)}{2f'(x_n)} e_n^2.$$

Therefore, from Eq. (2),

$$e_{n+1} = \frac{f''(\xi)}{2f'(x_n)} e_n^2.$$

Since $f \in C^2[a,b]$ and $f'(x_*) \neq 0$, $\left|\frac{f''(\xi)}{2f'(x_n)}\right| < C$ for some constant C in $[x_* - \delta, x_* + \delta]$, for some small $\delta > 0$.

When the initial guess x_0 is very close to x_* in the sense that $x_0 \in [x_* - \delta, x_* + \delta]$ with a small $\delta > 0$, such that

$$C|e_0| \leq \frac{1}{2}.$$

We have

$$|e_1| \leq Ce_0^2 \leq \frac{1}{2}|e_0|,$$

and accordingly,

$$|e_1| \leq |e_0| \leq \delta,$$

i.e. $x_1 \in [x_* - \delta, x_* + \delta]$.

Similarly, by mathematical induction, we can show that

$$|e_{n+1}| \leq \frac{1}{2}|e_n|,$$

and $x_{n+1} \in [x_* - \delta, x_* + \delta]$ for all n.

It can be calculated that

$$|e_1| \le \frac{1}{2}|e_0|$$

$$|e_2| \le \frac{1}{2}|e_1| \le \left(\frac{1}{2}\right)^2 |e_0|$$

$$\cdots$$

$$|e_n| \le \left(\frac{1}{2}\right)^n |e_0|.$$

Therefore, we have

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

Theorem. Let $f \in C^2[a,b]$. If $x_* \in (a,b)$ is such that $f(x_*) = 0$ and $f'(x_*) \neq 0$, then there exists a $\delta > 0$ such that Newton's method generates a sequence $\{x_n\}_{n=1}^{\infty}$ converges to x_* for any initial approximation $x_0 \in [x_* - \delta, x_* + \delta]$.

Denote the error $e_n = x_n - x_*$.

Newton's method gives $e_{n+1} = \dfrac{f''(\xi)}{2f'(x_n)} e_n^2.$