National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Numerical Computing (NC-2008)

Course Instructor

USAMA ANTULEY

usama.antuley@nu.edu.pk

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Lecture # 01

Orientation

About Course , Marking Division (Proposed) , Class Protocols

Let's Begin

Course Details

Textbook(s)	**Title**	Numerical Analysis , 9th Edition
	Author	Burden and Faires
	Publisher	BOOKS/COLE (Cengage Learning)
Ref. Book(s)	**Title**	Numerical Methods in Engineering with Python 3
	Author	Jaan Kiusalaas
	Title	Applied Numerical Methods with Matlab for Engineers and Scientist, 3rd Edition
	Author	Steven C,Chapra

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

MARKING DIVISION

Particulars	% Marks
Sessional I (Theory + Lab)	15
Sessional II (Theory + Lab)	15
Assignment (Theory + Lab)	10
Quiz/Lab Task	10
Final (Theory + Lab)	50
Total	**100**

For Complete Copy = 2

Protocols

- Be in Classroom on time
- Student who arrive more than 5 minutes late will be marked LATE & after 15 minutes as ABSENT
- Keep remember to turn off your Cell phone before entering the class
- Avoid conversation during lecture
- Submit your Assignment on time. **No submission after the deadline**
- Always bring your **Work Book/Note Book** and **Calculator** with you in the class

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Academic Calendar

Spring 2023 Semester

S. No	Week	Descriptions	Date
1	0	The registration process of the course (s)	Jan 18-23 (Wed - Mon)
2	1	Commencement Of the Classes	Jan 23 (Mon)
3	2	Add & Drop Of Courses	Feb 04 (Sat)
4	2	Last Date for Applying Semester's freeze	Feb 04 (Sat)
5	**6**	**Sessional-I Examinations**	**Feb 25 – Mar 01 (Sat - Wed)**
6	7	Procom Days	Mar 09 – Mar 10 (Thu – Fri)
7	**11**	**Sessional-II Examinations**	**Apr 05 – Apr 08 (Wed - Sat)**
8	15	Developers Day	May 04 (Thu)
9	16	Last Day of Classes	May 12 (Fri)
10	17	Last Date Of Withdrawal of Courses	May 19 (Fri)
11	**18-20**	**Final Examinations**	**May 22 - June 10 (Mon - Sat)**
12	22	Result Announcement	June 17 (Sat)

Google Classroom Code

ah7syr7

Steps:

1. Go to **classroom.google.com** and click Sign In. Sign in with your **NU ID**.
2. At the top, click Join class .
3. Enter the above-mentioned class code and click Join.

National University
of computer and emerging sciences

Foundation for Advancement of Science and Technology

Why Numerical Methods??

To accurately *approximate* the solutions of problems that cannot be solved exactly (by analytical method).

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Application of Numerical Computing in your domain??

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Some of the Applications:

- ✓ Image Processing
- ✓ Computer Vision
- ✓ Computer Graphics (rendering, animation),
- ✓ Climate Modeling,
- ✓ Weather Predictions,
- ✓ “Virtual” crash-testing of cars etc.
- ✓ medical imaging (CT = Computed Tomography),
- ✓ CAD (Computer-Aided Design)
- ✓ And many more

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Some Challenges/Issues in NC:

- ✓ Accuracy
- ✓ Precision
- ✓ Errors (True & Approximate)
- ✓ Significant Figures etc.

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

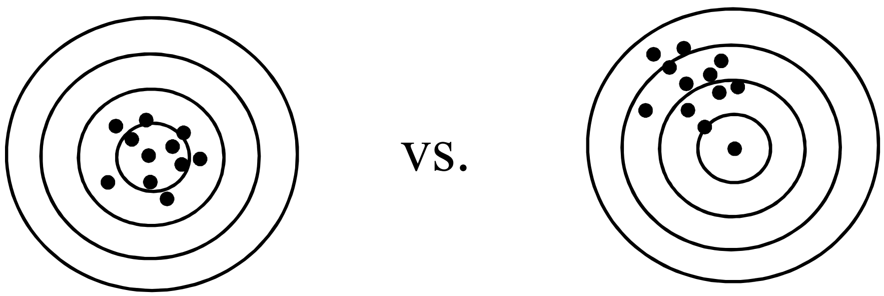
Accuracy:

- **Accuracy:**

  "How closely a computed value agrees with the true value"

- **Bias/Inaccuracy :**

  "A systematic deviation from the truth"



Accurate

Biased/Inaccurate

National University of computer and emerging sciences

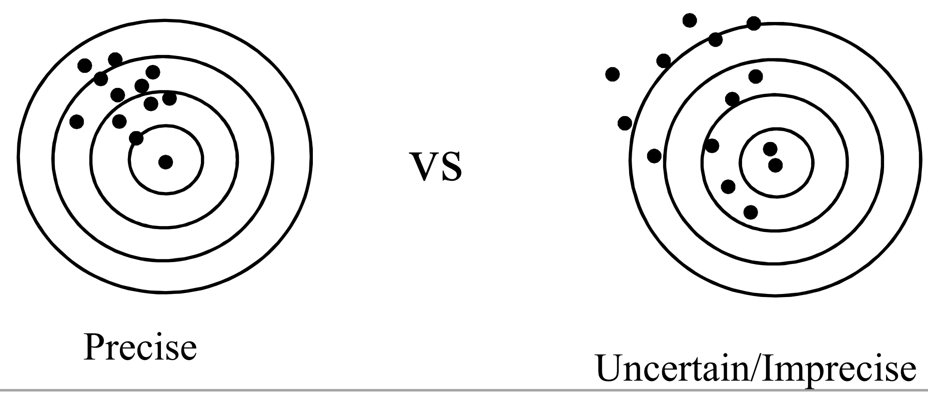
Foundation for Advancement of Science and Technology

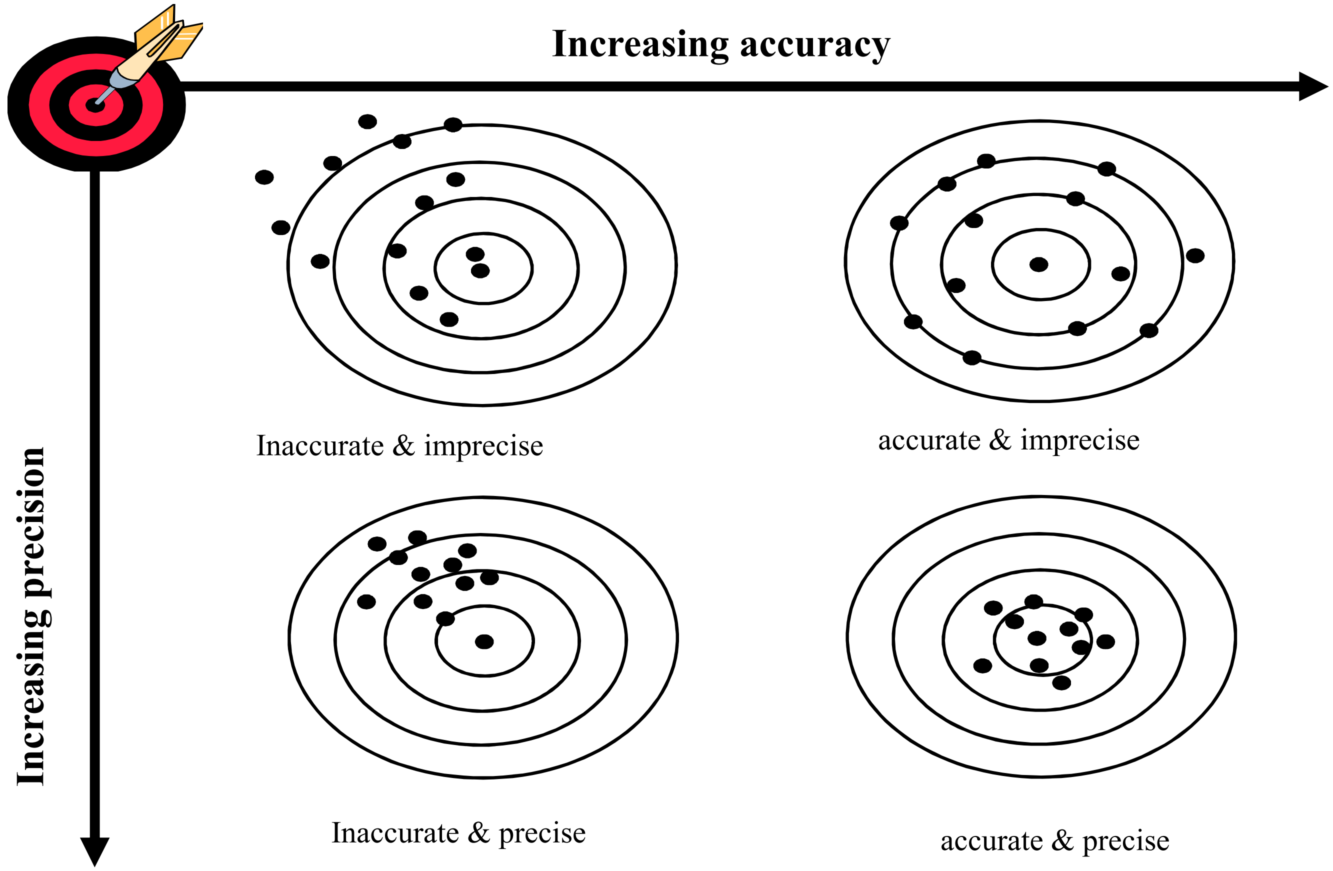
Precision:

- **Precision :**

  "How closely individual computed values agree with each other"
- **Uncertainty/Imprecision :**

  "magnitude of scatter"




Increasing accuracy
Increasing precision
Inaccurate & imprecise
accurate & imprecise
Inaccurate & precise
accurate & precise

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Measurement of Errors :

- When the current solution is compared with the true solution, the error involved is called **true error**
- When the current solution is compared with the solution obtained in the previous iteration, the error involved is called **approximate error**

True Errors :

"It is used to measure the lack of accuracy of an estimate"

- True (absolute) error = $\boxed{\mathbf{E_t} = \text{True value} - \text{Approximation}}$
- True Relative error = $\dfrac{E_t}{True\ value}$
- True Percent Relative Error $= \dfrac{True\,value - approximation}{True\,value} \times 100\%$

Approximate Errors :

"Used to measure the lack of precision of an estimate"

- **Approximate (Absolute) Error**

$\mathbf{E_a}$ = Current approximation - Previous approximation

- **Approximate Relative Error** = $\mathbf{E_a}$ / Current approx.

- **Approx. Percent Relative Error** $= \dfrac{\text{Current approx.} - \text{Previous approx}}{\text{Current approx}} \times 100\%$

Practice Problem:

The following sequence of estimates was obtained when a numerical method was applied to solve the equation:

$$x^4 - 5x - 7 = 0$$

1.8254 1.9633 2.0121 2.0283 2.0335 2.0351 2.0356 2.0358

Calculate the four errors for these estimates, given that one of the roots of the equation is 2.0359.

Lecture # 02

TYPES & MEASUREMENT OF ERRORS

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

REVIEW OF LAST CLASS:

Under Error Measurement, We discussed,

i. **Absolute(True) Error**

ii. **Relative & Percentage Relative Error**

Note:

The relative error is generally a better measure of accuracy than the absolute error because it takes into consideration the size of the number being approximated.

Example:

For example, suppose that you have the task of measuring the lengths of a bridge and a rivet and come up with 9999 and 9 cm, respectively. If the true values are 10,000 and 10 cm, respectively, the error in both cases is 1 cm. However, their percent relative errors can be computed using Eq. (4.3) as 0.01% and 10%, respectively. Thus, although both measurements have an absolute error of 1 cm, the relative error for the rivet is much greater. We would probably conclude that we have done an adequate job of measuring the bridge, whereas our estimate for the rivet leaves something to be desired.

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Approximate Errors :

$$\varepsilon_a = \frac{\text{present approximation} - \text{previous approximation}}{\text{present approximation}} 100\%$$

For iterative approximations, continue to iterate until the relative approximate error magnitude is less than a specified ***stopping criterion***

$$\boxed{|\varepsilon_a| < \varepsilon_s}$$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

For accuracy to ***at least n significant figures*** set the stopping criterion to

$$\boxed{\varepsilon_s = (0.5 \times 10^{2-n})\%}$$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Types of Errors :

Errors are of two types:

1. Truncation Error
2. Round-off Error

1. **Truncation error** is a result of using approximations to represent exact mathematical procedures ***OR*** *when an iterative method is terminated*

2. **Round-off error** occurs when only certain digits and decimal places are used to represent exact numbers.

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Truncation Error:

$$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$$

HINT:

9.8.1 DEFINITION If f has derivatives of all orders at x_0, then we call the series

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!}(x-x_0)^k = f(x_0) + f'(x_0)(x-x_0) + \frac{f''(x_0)}{2!}(x-x_0)^2 + \cdots + \frac{f^{(k)}(x_0)}{k!}(x-x_0)^k + \cdots \quad (1)$$

the ***Taylor series for f about $x = x_0$***. In the special case where $x_0 = 0$, this series becomes

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!}x^k = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \cdots + \frac{f^{(k)}(0)}{k!}x^k + \cdots \quad (2)$$

in which case we call it the ***Maclaurin series for f***.

$$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$$

Starting with the simplest version, $e^x = 1$, add terms one at a time in order to estimate $e^{0.5}$. After each new term is added, compute the true and approximate percent relative errors

Solution. First to determine the error criterion that ensures a result that is correct to at least three significant figures:

$$\varepsilon_s = (0.5 \times 10^{2-3})\% = 0.05\%$$

Thus, we will add terms to the series until ε_a falls below this level.

$$e^x = 1 + x$$

or for $x = 0.5$

$$e^{0.5} = 1 + 0.5 = 1.5$$

This represents a true percent relative error

$$\varepsilon_t = \left|\frac{1.648721 - 1.5}{1.648721}\right| \times 100\% = 9.02\%$$

to determine an approximate estimate of the error, as in

$$\varepsilon_a = \left|\frac{1.5 - 1}{1.5}\right| \times 100\% = 33.3\%$$

Because ε_a is not less than the required value of ε_s, we would continue the computation by adding another term, $x^2/2!$, and repeating the error calculations. The process is continued until $|\varepsilon_a| < \varepsilon_s$. The entire computation can be summarized as

Terms	Result	ε_t, %	ε_a, %
1	1	39.3	
2	1.5	9.02	33.3
3	1.625	1.44	7.69
4	1.645833333	0.175	1.27
5	1.648437500	0.0172	0.158
6	1.648697917	0.00142	0.0158

Representation of Real Numbers:

1. Binary Machine Numbers:

A 64-bit (binary digit) representation is used for a real number (according to IEEE standards).

$$(-1)^s 2^{c-1023}(1+f)$$

This representation is called floating point representation.

The first bit is a sign indicator, denoted s. This is followed by an 11-bit exponent, c, called the **characteristic**, and a 52-bit binary fraction, f, called the **mantissa**. The base for the exponent is 2.

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Examples:

0 10000000011 1011100100010000000000000000000000000000000000000000

Sign Characteristic Mantissa

Sign: (0: positive; 1:negative)

$$c = 1 \cdot 2^{10} + 0 \cdot 2^{9} + \cdots + 0 \cdot 2^{2} + 1 \cdot 2^{1} + 1 \cdot 2^{0} = 1024 + 2 + 1 = 1027.$$

The exponential part of the number is, therefore:

$$2^{1027-1023} = 2^{4}$$

Mantissa The final 52 bits is:

$$f = 1 \times \left(\frac{1}{2}\right)^1 + 1 \times \left(\frac{1}{2}\right)^3 + 1 \times \left(\frac{1}{2}\right)^4 + 1 \times \left(\frac{1}{2}\right)^5 + 1 \times \left(\frac{1}{2}\right)^8 + 1 \times \left(\frac{1}{2}\right)^{12}$$

As a consequence, this machine number precisely represents the decimal number

$$(-1)^s 2^{c-1023}(1+f) = (-1)^0 \cdot 2^{1027-1023}\left(1 + \left(\frac{1}{2} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \frac{1}{256} + \frac{1}{4096}\right)\right)$$

$$= 27.56640625.$$

Reading Activity: Page # 19 of Textbook

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Overflow & Underflow

An overflow error is produced when trying to use a number too large (greater than the corresponding R_{max}):

- In most computers, execution is aborted.
- IEEE format may support them by assigning the symbolic values

$$\pm\infty \quad \text{or} \quad NaN.$$

An underflow error is produced when trying to use a number too small (less, in absolute value, than the corresponding R_{min}). Two possible behaviors:

- It lies in the range of denormalized numbers, so it is still representable. In this case, precision decreases and it is called gradual underflow.
- Otherwise, it is identified to 0.

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

2. Decimal Machine Numbers: (Normalized Floating Point Representation)

$$\boxed{\pm 0.d_1 d_2 \dots d_k \times 10^n}, \quad 1 \le d_1 \le 9, \quad 0 \le d_i \le 9$$

Any positive real number within the numerical range of the machine can be normalized to the form:

$$y = 0.d_1 d_2 \dots d_k d_{k+1} d_{k+2} \dots \times 10^n$$

The floating-point form of y, denoted $fl(y)$, is obtained by terminating the mantissa of y at k decimal digits. This can be performed by using one of two methods:

1. **Chopping:**

$$fl(y) = 0.d_1d_2 \dots d_k \times 10^n$$

2. **Rounding:**

$$fl(y) = 0.\delta_1\delta_2 \dots \delta_k \times 10^n$$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Example: Convert the following numbers to 4-digit by chopping and rounding:

$$x = 635894 \;,\; y = 0.00218 \;,\; z = 584.63$$

Chopping:

$$x^* = 0.6358 \times 10^6$$

Rounding:

$$x^* = 0.6359 \times 10^6$$

Similarly do for y & z

Chopping: $y^* = 0.2180 \times 10^{-2}, \quad z^* = 0.5486 \times 10^{3}$

Rounding: $y^* = 0.2180 \times 10^{-2}, \quad z^* = 0.5486 \times 10^{3}$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Definition 1: Suppose that p^* is an approximation to p. The **absolute error** is $e_p = |p - p^*|$, and the **relative error** is $\delta_p = \frac{|p-p^*|}{|p|}$ provided that $p \neq 0$.

Determine the absolute and relative errors when approximating p by p^* when

(a) $p = 0.3000 \times 10^1$ and $p^* = 0.3100 \times 10^1$;

(b) $p = 0.3000 \times 10^{-3}$ and $p^* = 0.3100 \times 10^{-3}$;

(c) $p = 0.3000 \times 10^4$ and $p^* = 0.3100 \times 10^4$.

Solution

(a) For $p = 0.3000 \times 10^1$ and $p^* = 0.3100 \times 10^1$ the absolute error is 0.1, and the relative error is $0.333\overline{3} \times 10^{-1}$.

(b) $p = 0.3000 \times 10^{-3}$ and $p^* = 0.3100 \times 10^{-3}$;

(c) $p = 0.3000 \times 10^{4}$ and $p^* = 0.3100 \times 10^{4}$.

(b) For $p = 0.3000 \times 10^{-3}$ and $p^* = 0.3100 \times 10^{-3}$ the absolute error is 0.1×10^{-4}, and the relative error is $0.333\overline{3} \times 10^{-1}$.

(c) For $p = 0.3000 \times 10^{4}$ and $p^* = 0.3100 \times 10^{4}$, the absolute error is 0.1×10^{3}, and the relative error is again $0.333\overline{3} \times 10^{-1}$.

Lecture # 03

Finite Digit & Nested Arithmetic

National University
of computer and emerging sciences

Foundation for Advancement of Science and Technology FAST

Finite Digit Arithmetic:

Example 3 Suppose that $x = \frac{5}{7}$ and $y = \frac{1}{3}$. Use five-digit chopping for calculating $x + y$, $x - y$, $x \times y$, and $x \div y$.

$$x = \frac{5}{7} = 0.\overline{714285} \quad \text{and} \quad y = \frac{1}{3} = 0.\overline{3}$$

$$x \oplus y = fl(fl(x) + fl(y)) = fl\left(0.71428 \times 10^{0} + 0.33333 \times 10^{0}\right)$$

$$= fl\left(1.04761 \times 10^{0}\right) = 0.10476 \times 10^{1}.$$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Error Analysis:

The true value is $x + y = \frac{5}{7} + \frac{1}{3} = \frac{22}{21}$, so we have

$$\text{Absolute Error} = \left| \frac{22}{21} - 0.10476 \times 10^1 \right| = 0.190 \times 10^{-4}$$

and

$$\text{Relative Error} = \left| \frac{0.190 \times 10^{-4}}{22/21} \right| = 0.182 \times 10^{-4}.$$

Operation	Result	Actual value	Absolute error	Relative error
$x \oplus y$	0.10476×10^1	$22/21$	0.190×10^{-4}	0.182×10^{-4}
$x \ominus y$	0.38095×10^0	$8/21$	0.238×10^{-5}	0.625×10^{-5}
$x \otimes y$	0.23809×10^0	$5/21$	0.524×10^{-5}	0.220×10^{-4}
$x \oplus\!\!\!\!\div y$	0.21428×10^1	$15/7$	0.571×10^{-4}	0.267×10^{-4}

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Loss of Significance:

Let $p = 0.54617$ and $q = 0.54601$. Use four-digit arithmetic to approximate $p - q$ and determine the absolute and relative errors using **(a)** rounding and **(b)** chopping.

Solution The exact value of $r = p - q$ is $r = 0.00016$.

(a) Suppose the subtraction is performed using four-digit rounding arithmetic. Rounding p and q to four digits gives $p^* = 0.5462$ and $q^* = 0.5460$, respectively, and $r^* = p^* - q^* = 0.0002$ is the four-digit approximation to r. Since

$$\frac{|r - r^*|}{|r|} = \frac{|0.00016 - 0.0002|}{|0.00016|} = 0.25,$$

the result has only one significant digit, whereas p^* and q^* were accurate to four and five significant digits, respectively.

(b) If chopping is used to obtain the four digits, the four-digit approximations to p, q, and r are $p^* = 0.5461$, $q^* = 0.5460$, and $r^* = p^* - q^* = 0.0001$. This gives

$$\frac{|r - r^*|}{|r|} = \frac{|0.00016 - 0.0001|}{|0.00016|} = 0.375,$$

which also results in only one significant digit of accuracy. ■

National University
of computer and emerging sciences

Foundation for Advancement of Science and Technology

Loss of Significance:

occurs in numerical calculations when too many significant digits cancel

$$\begin{array}{r} 123.4567 \\ -\ 123.4566 \\ \hline 000.0001 \end{array}$$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Remedy:

Example

Calculate $\sqrt{9.01} - 3$ on a three-decimal-digit

$$
\begin{aligned}
\sqrt{9.01} - 3 &= \frac{(\sqrt{9.01} - 3)(\sqrt{9.01} + 3)}{\sqrt{9.01} + 3} \\
&= \frac{9.01 - 3^2}{\sqrt{9.01} + 3} \\
&= \frac{0.01}{3.00 + 3} = \frac{.01}{6} = 0.00167 \approx 1.67 \times 10^{-3}.
\end{aligned}
$$

National University
of computer and emerging sciences

Foundation for Advancement of Science and Technology

Remedies:

1. Rationalizing
2. Using Series Expansion
3. Using Trigonometric identities
4. Reformulation

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Remedies:

Using Trigonometric identities

As a simple example, consider the function

$$f(x) = \cos^2(x) - \sin^2(x)$$

There will be loss of significance at $x = \pi/4$.

The problem can be solved by the simple substitution

$$\cos^2(x) - \sin^2(x) = cos(2x)$$

The quadratic formula states that the roots of $ax^2 + bx + c = 0$, when $a \neq 0$, are

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{and} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}.$$

Consider this formula applied to the equation $x^2 + 62.10x + 1 = 0$, whose roots are approximately

$$x_1 = -0.01610723 \quad \text{and} \quad x_2 = -62.08390.$$

use four-digit rounding arithmetic in the calculations to determine the root

$$\begin{aligned}\sqrt{b^2 - 4ac} &= \sqrt{(62.10)^2 - (4.000)(1.000)(1.000)} \\ &= \sqrt{3856. - 4.000} = \sqrt{3852.} = 62.06,\end{aligned}$$

$$fl(x_1) = \frac{-62.10 + 62.06}{2.000} = \frac{-0.04000}{2.000} = -0.02000,$$

a poor approximation to $x_1 = -0.01611$, with the large relative error

$$\frac{|-0.01611 + 0.02000|}{|-0.01611|} \approx 2.4 \times 10^{-1}.$$

$$fl(x_2) = \frac{-62.10 - 62.06}{2.000} = \frac{-124.2}{2.000} = -62.10$$

has the small relative error

$$\frac{|-62.08 + 62.10|}{|-62.08|} \approx 3.2 \times 10^{-4}.$$

To overcome loss of significance of x_1

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}\left(\frac{-b - \sqrt{b^2 - 4ac}}{-b - \sqrt{b^2 - 4ac}}\right) = \frac{b^2 - (b^2 - 4ac)}{2a(-b - \sqrt{b^2 - 4ac})},$$

which simplifies to an alternate quadratic formula

$$x_1 = \frac{-2c}{b + \sqrt{b^2 - 4ac}}. \tag{1.2}$$

Using (1.2) gives

$$fl(x_1) = \frac{-2.000}{62.10 + 62.06} = \frac{-2.000}{124.2} = -0.01610,$$

which has the small relative error 6.2×10^{-4}.

Nested Arithmetic:

Accuracy loss due to round-off error can also be reduced by rearranging calculations.

Evaluate $f(x) = x^3 - 6.1x^2 + 3.2x + 1.5$ at $x = 4.71$ using three-digit arithmetic.

	x	x^2	x^3	$6.1x^2$	$3.2x$
Exact	4.71	22.1841	104.487111	135.32301	15.072
Three-digit (chopping)	4.71	22.1	104.	134.	15.0
Three-digit (rounding)	4.71	22.2	105.	135.	15.1

$$\text{Chopping: } \left| \frac{-14.263899 + 13.5}{-14.263899} \right| \approx 0.05, \quad \text{and Rounding: } \left| \frac{-14.263899 + 13.4}{-14.263899} \right| \approx 0.06. \quad \blacksquare$$

Nested Scheme for the same problem:

$$f(x) = x^3 - 6.1x^2 + 3.2x + 1.5 = ((x - 6.1)x + 3.2)x + 1.5.$$

Using three-digit chopping arithmetic now produces

$$\begin{aligned} f(4.71) &= ((4.71 - 6.1)4.71 + 3.2)4.71 + 1.5 = ((-1.39)(4.71) + 3.2)4.71 + 1.5 \\ &= (-6.54 + 3.2)4.71 + 1.5 = (-3.34)4.71 + 1.5 = -15.7 + 1.5 = -14.2. \end{aligned}$$

In a similar manner, we now obtain a three-digit rounding answer of -14.3. The new relative errors are

$$\text{Three-digit (chopping):} \quad \left|\frac{-14.263899 + 14.2}{-14.263899}\right| \approx 0.0045;$$

$$\text{Three-digit (rounding):} \quad \left|\frac{-14.263899 + 14.3}{-14.263899}\right| \approx 0.0025.$$

Nesting has reduced the relative error for the chopping approximation to less than 10% of that obtained initially. For the rounding approximation the improvement has been even more dramatic; the error in this case has been reduced by more than 95%. □

Previously:

$$\text{Chopping:} \left|\frac{-14.263899 + 13.5}{-14.263899}\right| \approx 0.05, \quad \text{and Rounding:} \left|\frac{-14.263899 + 13.4}{-14.263899}\right| \approx 0.06.$$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Do Q.1,4,5,6,7,8 & 13 from Ex # 1.2

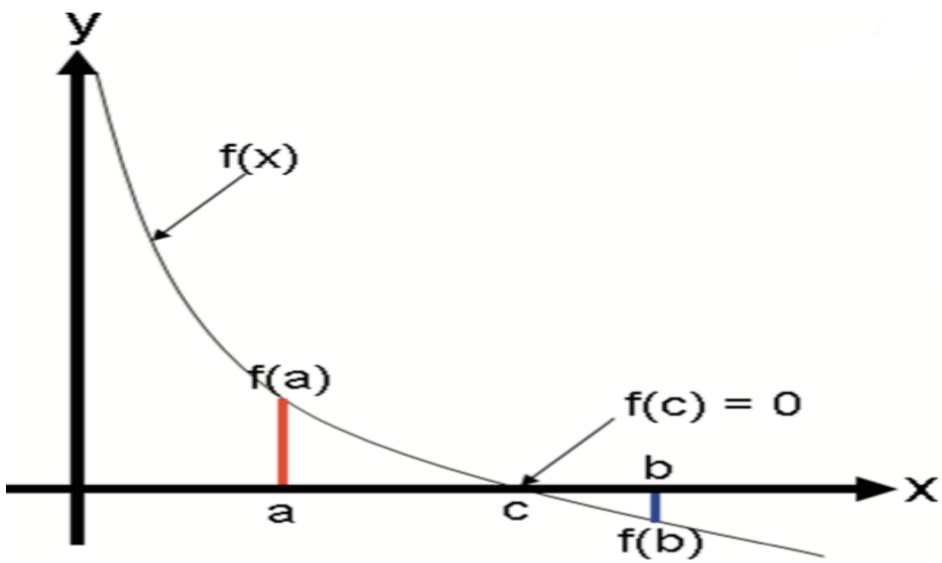
Lecture # 04

Root of equations in one variable

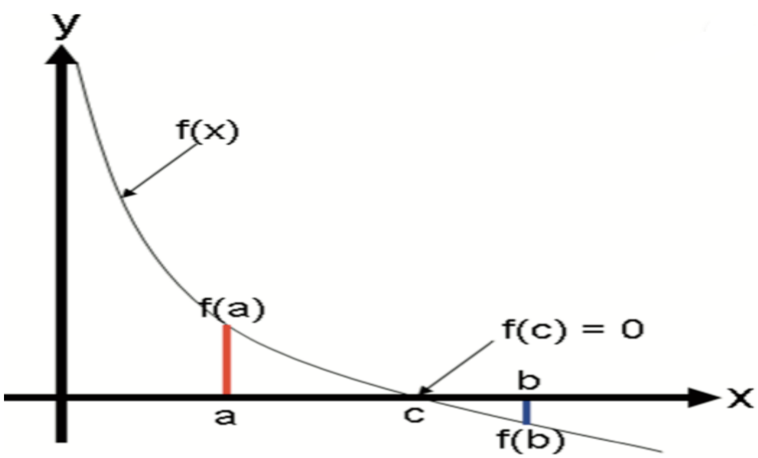
Bisection or Binary Search Method

Bisection OR Binary Search Method:

The **Intermediate Value Theorem** says that if $f(x)$ is a continuous function between a and b, and $\text{sign}(f(a)) \neq \text{sign}(f(b))$, then there must be a c, such that $a < c < b$ and $f(c) = 0$. This is illustrated in the following figure.



The **bisection method** uses the intermediate value theorem iteratively to find roots. Let $f(x)$ be a continuous function, and a and b be real scalar values such that $a < b$. Assume, without loss of generality, that $f(a) > 0$ and $f(b) < 0$. Then by the intermediate value theorem, there must be a root on the open interval (a, b). Now let $m = \frac{b+a}{2}$, the midpoint between and a and b. If $f(m) = 0$ or is close enough, then m is a root. If $f(m) > 0$, then m is an improvement on the left bound, a, and there is guaranteed to be a root on the open interval (m, b). If $f(m) < 0$, then m is an improvement on the right bound, b, and there is guaranteed to be a root on the open interval (a, m).



Bisection OR Binary Search Method:

The **bisection method** uses the intermediate value theorem iteratively to find roots.

In computer science, the process of dividing a set continually in half to search for the solution to a problem, as the bisection method does, is known as a *binary search* procedure.

Algorithm:

The steps to apply the bisection method to find the root of the equation $f(x) = 0$ are

1. Choose x_λ and x_u as two guesses for the root such that $f(x_\lambda)f(x_u) < 0$, or in other words, $f(x)$ changes sign between x_λ and x_u.
2. Estimate the root, x_m, of the equation $f(x) = 0$ as the mid-point between x_λ and x_u as

$$x_m = \frac{x_\lambda + x_u}{2}$$

Algorithm:

3. Now check the following
   a) If $f(x_\lambda)f(x_m) < 0$, then the root lies between x_λ and x_m; then $x_\lambda = x_\lambda$ and $x_u = x_m$.
   b) If $f(x_\lambda)f(x_m) > 0$, then the root lies between x_m and x_u; then $x_\lambda = x_m$ and $x_u = x_u$.
   c) If $f(x_\lambda)f(x_m) = 0$; then the root is x_m. Stop the algorithm if this is true.
4. Find the new estimate of the root

$$x_m = \frac{x_\lambda + x_u}{2}$$

Find the absolute relative approximate error as

$$|\in_a| = \left| \frac{x_m^{\text{new}} - x_m^{\text{old}}}{x_m^{\text{new}}} \right| \times 100$$

where

x_m^{new} = estimated root from present iteration

x_m^{old} = estimated root from previous iteration

Algorithm:

5. Compare the absolute relative approximate error $|\in_a|$ with the pre-specified relative error tolerance $\in_s$. If $|\in_a| > \in_s$, then go to Step 3, else stop the algorithm. Note one should also check whether the number of iterations is more than the maximum number of iterations allowed. If so, one needs to terminate the algorithm and notify the user about it.

Number of iterations for getting root:

$$n = \frac{\ln(\Delta x/\varepsilon)}{\ln 2}$$

Example:

Show that $f(x) = x^3 + 4x^2 - 10 = 0$ has a root in $[1, 2]$, and use the Bisection method to determine an approximation to the root that is accurate to at least within 10^{-4}.

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Solution Because $f(1) = -5$ and $f(2) = 14$ the Intermediate Value Theorem that this continuous function has a root in $[1, 2]$.

For the first iteration of the Bisection method we use the fact that at the midpoint of $[1, 2]$ we have $f(1.5) = 2.375 > 0$. This indicates that we should select the interval $[1, 1.5]$

n	a_n	b_n	p_n	$f(p_n)$
1	1.0	2.0	1.5	2.375
2	1.0	1.5	1.25	−1.79687
3	1.25	1.5	1.375	0.16211
4	1.25	1.375	1.3125	−0.84839
5	1.3125	1.375	1.34375	−0.35098
6	1.34375	1.375	1.359375	−0.09641
7	1.359375	1.375	1.3671875	0.03236
8	1.359375	1.3671875	1.36328125	−0.03215
9	1.36328125	1.3671875	1.365234375	0.000072
10	1.36328125	1.365234375	1.364257813	−0.01605
11	1.364257813	1.365234375	1.364746094	−0.00799
12	1.364746094	1.365234375	1.364990235	−0.00396
13	1.364990235	1.365234375	1.365112305	−0.00194

The Bisection method, though conceptually clear, has significant drawbacks. It is relatively slow to converge (that is, N may become quite large before $|p - p_N|$ is sufficiently small), and a good intermediate approximation might be inadvertently discarded. However, the method has the important property that it always converges to a solution, and for that reason it is often used as a starter for the more efficient methods we will see later in this chapter.

Example:

Determine the number of iterations necessary to solve $f(x) = x^3 + 4x^2 - 10 = 0$ with accuracy 10^{-4} using $a_1 = 1$ and $b_1 = 2$.

HINT:

$$n = \frac{\ln(\Delta x/\varepsilon)}{\ln 2}$$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Do Q 1,2,3,4,5,6,12 & 13 from Ex # 2.1

Lecture # 05

Root of equations in one variable

Fixed Point Iteration

Fixed Point Iteration:

The number p is a **fixed point** for a given function g if $g(p) = p$.

OR

$$x = \emptyset(x)$$

Algorithm:

To find a solution to $p = g(p)$ given an initial approximation p_0:

INPUT initial approximation p_0; tolerance TOL; maximum number of iterations N_0.

OUTPUT approximate solution p or message of failure.

Step 1 Set $i = 1$.

Step 2 While $i \le N_0$ do Steps 3–6.

Step 3 Set $p = g(p_0)$. (*Compute* p_i.)

Step 4 If $|p - p_0| < TOL$ then
OUTPUT (p); (*The procedure was successful.*)
STOP.

Step 5 Set $i = i + 1$.

Step 6 Set $p_0 = p$. (*Update* p_0.)

Step 7 OUTPUT ('The method failed after N_0 iterations, N_0 =', N_0);
(*The procedure was unsuccessful.*)
STOP.

Algorithm:

Iteration Algorithm with the Form $x = g(x)$

To determine a root of $f(x) = 0$, given a value x_1 reasonably close to the root,

Rearrange the equation to an equivalent form $x = g(x)$.

Repeat
Set $x_2 = x_1$.
Set $x_1 = g(x_1)$
Until $|x_1 - x_2| <$ tolerance value

Note: The method may converge to a root different from the expected one, or it may diverge. Different rearrangements will converge at different rates.

Example

- Fixed Point Iteration

$$f(x) = x^2 - 2x - 3 = 0$$

(ans: $x = 3$ or -1)

Case a:

$$\boxed{\begin{aligned} & x^2 - 2x - 3 = 0 \\ & \Rightarrow x^2 = 2x + 3 \\ & \Rightarrow x = \sqrt{2x+3} \\ & \Rightarrow g(x) = \sqrt{2x+3} \end{aligned}}$$

Case b:

$$\boxed{\begin{aligned} & x^2 - 2x - 3 = 0 \\ & \Rightarrow x(x-2) - 3 = 0 \\ & \Rightarrow x = \frac{3}{x-2} \\ & \Rightarrow g(x) = \frac{3}{x-2} \end{aligned}}$$

Case c:

$$\boxed{\begin{aligned} & x^2 - 2x - 3 = 0 \\ & \Rightarrow 2x = x^2 - 3 \\ & \Rightarrow x = \frac{x^2-3}{2} \\ & \Rightarrow g(x) = \frac{x^2-3}{2} \end{aligned}}$$

So which one is better?

Case a

$$x_{i+1} = \sqrt{2x_i + 3}$$

1.	$x_0 = 4$
2.	$x_1 = 3.31662$
3.	$x_2 = 3.10375$
4.	$x_3 = 3.03439$
5.	$x_4 = 3.01144$
6.	$x_5 = 3.00381$

Converge!

Case b

$$x_{i+1} = \frac{3}{x_i - 2}$$

1.	$x_0 = 4$
2.	$x_1 = 1.5$
3.	$x_2 = -6$
4.	$x_3 = -0.375$
5.	$x_4 = -1.263158$
6.	$x_5 = -0.919355$
7.	$x_6 = -1.02762$
8.	$x_7 = -0.990876$
9.	$x_8 = -1.00305$

Converge, but slower

Case c

$$x_{i+1} = \frac{x_i^2 - 3}{2}$$

1.	$x_0 = 4$
2.	$x_1 = 6.5$
3.	$x_2 = 19.625$
4.	$x_3 = 191.070$

Diverge!

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

(Fixed-Point Theorem)

Let $g \in C[a, b]$ be such that $g(x) \in [a, b]$, for all x in $[a, b]$. Suppose, in addition, that g' exists on (a, b) and that a constant $0 < k < 1$ exists with

$$|g'(x)| \leq k, \quad \text{for all } x \in (a, b).$$

Then for any number p_0 in $[a, b]$, the sequence defined by

$$p_n = g(p_{n-1}), \quad n \geq 1,$$

converges to the unique fixed point p in $[a, b]$. ■

Do Q 1,2,3,4,5,6,9,10,11 & 14 from Ex # 2.2

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Do Class Question

H.W:

Find the root of the transcendental equation $\cos x - 3x + 1 = 0$ correct up to seven decimal values.

Ans: 0.6071016

National University
of computer and emerging sciences

Foundation for Advancement
of Science and Technology

Lecture # 06

Newton's Raphson and Secant Method

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Newton's Raphson Method:

The Newton-Raphson algorithm is the **best-known** method of finding roots for a good reason:

- **It is simple and fast**.
- The only drawback of the method is that it uses the derivative $f'(x)$ of the function as well as the function $f(x)$ itself. Therefore, the Newton-Raphson method is usable only in problems where $f'(x)$ can be readily computed.

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

History:

Isaac Newton (1641–1727) was one of the most brilliant scientists of all time. The late 17th century was a vibrant period for science and mathematics and Newton's work touched nearly every aspect of mathematics. His method for solving was introduced to find a root of the equation $y^3 - 2y - 5 = 0$. Although he demonstrated the method only for polynomials, it is clear that he realized its broader applications.

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Formula:

$$p \approx p_0 - \frac{f(p_0)}{f'(p_0)} \equiv p_1.$$

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}, \quad \text{for } n \geq 1.$$

Algorithm:

To find a solution to $f(x) = 0$ given an initial approximation p_0:

INPUT initial approximation p_0; tolerance TOL; maximum number of iterations N_0.

OUTPUT approximate solution p or message of failure.

Step 1 Set $i = 1$.

Step 2 While $i \leq N_0$ do Steps 3–6.

Step 3 Set $p = p_0 - f(p_0)/f'(p_0)$. (*Compute p_i.*)

Step 4 If $|p - p_0| < TOL$ then
OUTPUT (p); (*The procedure was successful.*)
STOP.

Step 5 Set $i = i + 1$.

Step 6 Set $p_0 = p$. (*Update p_0.*)

Step 7 OUTPUT ('The method failed after N_0 iterations, $N_0 =$', N_0);
(*The procedure was unsuccessful.*)
STOP.

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Example:

Consider the function $f(x) = \cos x - x = 0$. Approximate a root of f using **(a)** a fixed-point method, and **(b)** Newton's Method

Note that the variable in the trigonometric function is in radian measure, not degrees. This will always be the case unless specified otherwise.

n	p_n
0	0.7853981635
1	0.7071067810
2	0.7602445972
3	0.7246674808
4	0.7487198858
5	0.7325608446
6	0.7434642113
7	0.7361282565

Table 2.3 shows the results of fixed-point iteration with $p_0 = \pi/4$. The best we could conclude from these results is that $p \approx 0.74$.

(b) To apply Newton's method to this problem we need $f'(x) = -\sin x - 1$. Starting again with $p_0 = \pi/4$, we generate the sequence defined, for $n \geq 1$, by

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f(p'_{n-1})} = p_{n-1} - \frac{\cos p_{n-1} - p_{n-1}}{-\sin p_{n-1} - 1}.$$

Newton's Method

n	p_n
0	0.7853981635
1	0.7395361337
2	0.7390851781
3	0.7390851332
4	0.7390851332

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Q: Find the root of $x^3 - 3x - 5 = 0$ $\boldsymbol{using\ Newton's\ Raphson\ Method}$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Q: Find the root of $x^3 - 3x - 5 = 0$ $\boldsymbol{using\ Newton's\ Raphson\ Method}$

Ans: $\boldsymbol{x_4 = 2.2790}$

Q: Find the root of $\sin x = 1 + x^3$ $\boldsymbol{using\ Newton's\ Raphson\ Method}$

Ans: $\boldsymbol{x_6 = -1.2490522}$

National University
of computer and emerging sciences

Foundation for Advancement of Science and Technology

Lecture # 07

Secant Method

National University of computer and emerging sciences

Foundation for Advancement FAST

Secant Method:

To find a solution to $f(x) = 0$ given initial approximations p_0 and p_1:

INPUT initial approximations p_0, p_1; tolerance TOL; maximum number of iteratio

OUTPUT approximate solution p or message of failure.

Step 1 Set $i = 2$;
$q_0 = f(p_0)$;
$q_1 = f(p_1)$.

Step 2 While $i \leq N_0$ do Steps 3–6.

Step 3 Set $p = p_1 - q_1(p_1 - p_0)/(q_1 - q_0)$. (*Compute* p_i.)

Step 4 If $|p - p_1| < TOL$ then
OUTPUT (p); (*The procedure was successful.*)
STOP.

Step 5 Set $i = i + 1$.

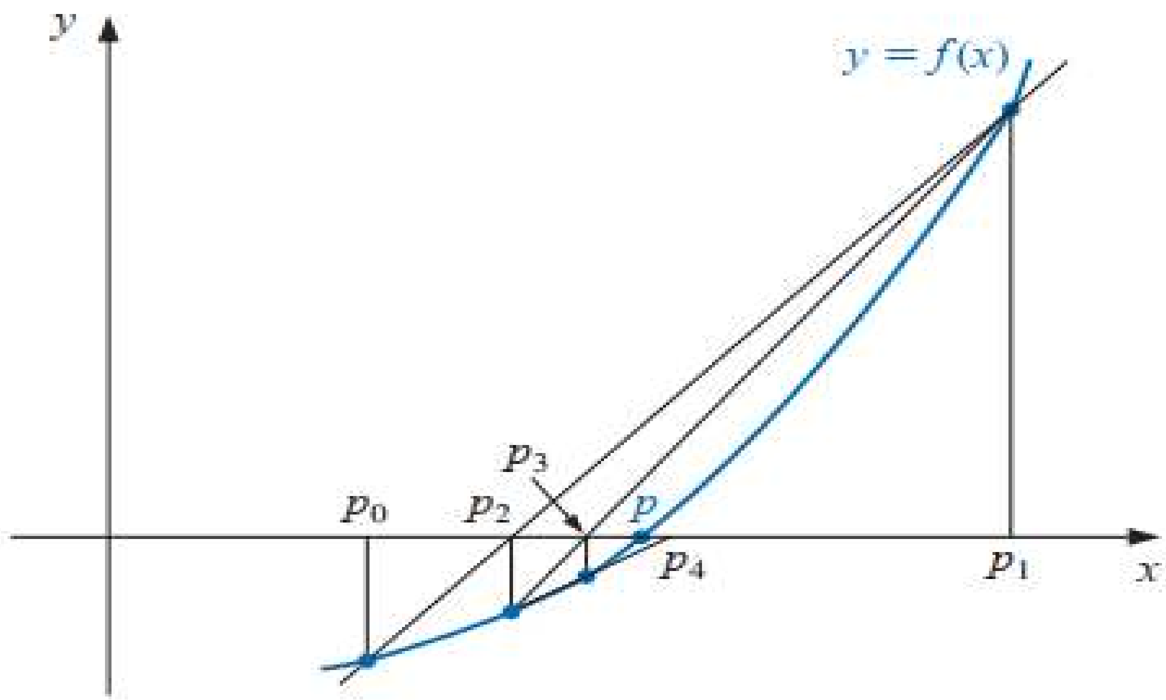
Step 6 Set $p_0 = p_1$; (*Update* p_0, q_0, p_1, q_1.)
$q_0 = q_1$;
$p_1 = p$;
$q_1 = f(p)$.

Step 7 OUTPUT ('The method failed after N_0 iterations, $N_0 =$', N_0);
(*The procedure was unsuccessful.*)
STOP.

Secant Method:

The word secant is derived from the Latin word *secan*, which means to cut. The secant method uses a secant line, a line joining two points that cut the curve, to approximate a root.

Secant Method:



Use the Secant method to find a solution to $x = \cos x$, and compare the approximations with those given in Example 1 which applied Newton's method.

n	p_n
0	0.7853981635
1	0.7071067810
2	0.7602445972
3	0.7246674808
4	0.7487198858
5	0.7325608446
6	0.7434642113
7	0.7361282565

	Newton
n	p_n
0	0.7853981635
1	0.7395361337
2	0.7390851781
3	0.7390851332
4	0.7390851332

	Secant
n	p_n
0	0.5
1	0.7853981635
2	0.7363841388
3	0.7390581392
4	0.7390851493
5	0.7390851332

Lecture # 09

False Position

(Regula Falsi or Linear Interpolation Method)

National University
of computer and emerging sciences

Foundation for Advancement of Science and Technology

Root Finding Methods:

- *Bracketing methods*. As the name implies, these are based on two initial guesses that "bracket" the root—that is, are on either side of the root.
- *Open methods*. These methods can involve one or more initial guesses, but there is no need for them to bracket the root.

False Position Method:

The **method of False Position** (also called *Regula Falsi*) generates approximations in the same manner as the Secant method, but it includes a test to ensure that the root is always bracketed between successive iterations

The term *Regula Falsi*, literally a false rule or false position, refers to a technique that uses results that are known to be false, but in some specific manner, to obtain convergence to a true result. False position problems can be found on the Rhind papyrus, which dates from about 1650 B.C.E.

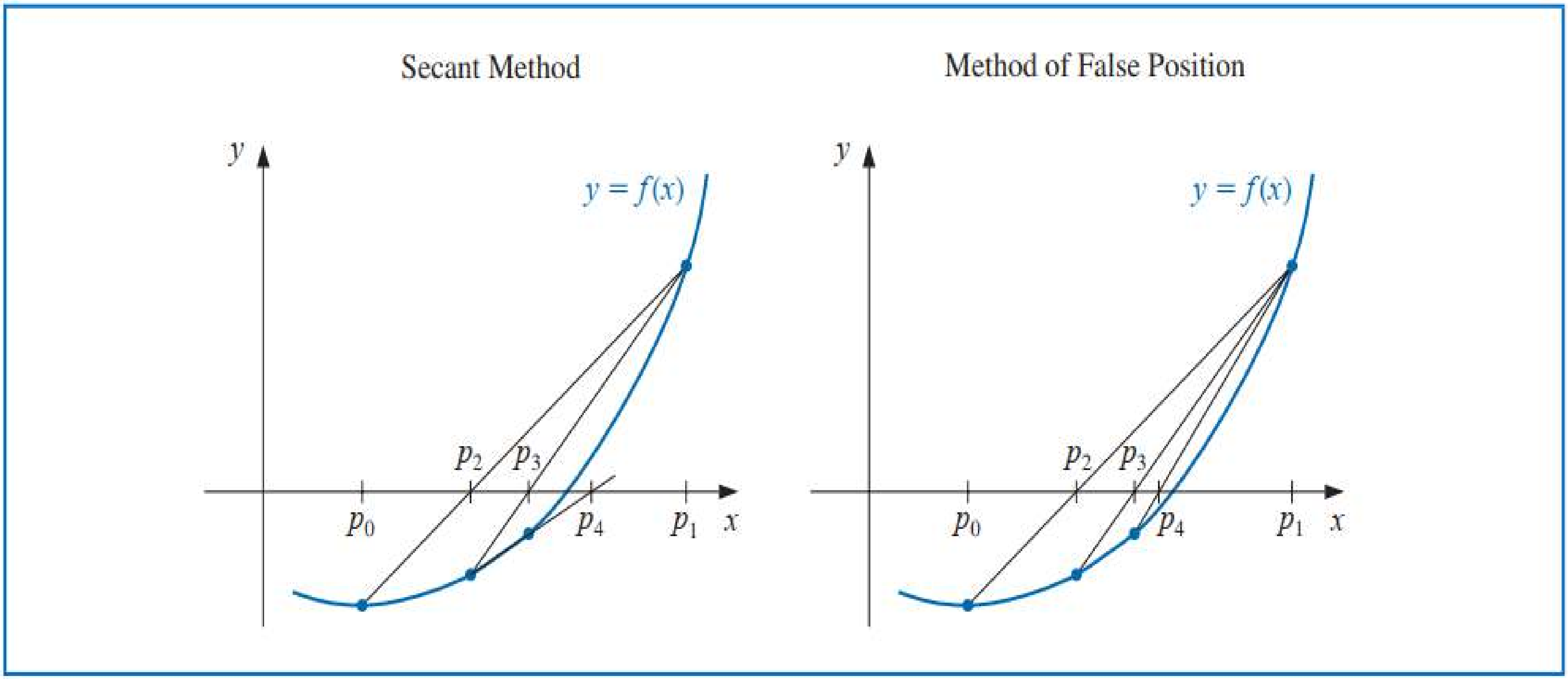
National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Working/Iteration Rule:

First choose initial approximations p_0 and p_1 with $f(p_0) \cdot f(p_1) < 0$. The approximation p_2 is chosen in the same manner as in the Secant method, as the x-intercept of the line joining $(p_0, f(p_0))$ and $(p_1, f(p_1))$. To decide which secant line to use to compute p_3, consider $f(p_2) \cdot f(p_1)$, or more correctly $\text{sgn}\, f(p_2) \cdot \text{sgn}\, f(p_1)$.

- If $\text{sgn}\, f(p_2) \cdot \text{sgn}\, f(p_1) < 0$, then p_1 and p_2 bracket a root. Choose p_3 as the x-intercept of the line joining $(p_1, f(p_1))$ and $(p_2, f(p_2))$.
- If not, choose p_3 as the x-intercept of the line joining $(p_0, f(p_0))$ and $(p_2, f(p_2))$, and then interchange the indices on p_0 and p_1.


Secant Method
Method of False Position
y
x
y = f(x)
p0
p1
p2
p3
p4

Algorithm:

To find a solution to $f(x) = 0$ given the continuous function f on the interval $[p_0, p_1]$ where $f(p_0)$ and $f(p_1)$ have opposite signs:

INPUT initial approximations p_0, p_1; tolerance TOL; maximum number of iterations N_0.

OUTPUT approximate solution p or message of failure.

Step 1 Set $i = 2$;
$q_0 = f(p_0)$;
$q_1 = f(p_1)$.

Step 2 While $i \le N_0$ do Steps 3–7.

Step 3 Set $p = p_1 - q_1(p_1 - p_0)/(q_1 - q_0)$. *(Compute p_i.)*

Step 4 If $|p - p_1| < TOL$ then
OUTPUT (p); *(The procedure was successful.)*
STOP.

Step 5 Set $i = i + 1$;
$q = f(p)$.

Step 6 If $q \cdot q_1 < 0$ then set $p_0 = p_1$;
$q_0 = q_1$.

Step 7 Set $p_1 = p$;
$q_1 = q$.

Step 8 OUTPUT ('Method failed after N_0 iterations, $N_0 =$', N_0);
(The procedure unsuccessful.)
STOP.

Example: Find the solution $x = cosx$

Solution To make a reasonable comparison we will use the same initial approximations as in the Secant method, that is, $p_0 = 0.5$ and $p_1 = \pi/4$. Table 2.6 shows the results of the method of False Position applied to $f(x) = \cos x - x$ together with those we obtained using the Secant and Newton's methods. Notice that the False Position and Secant approximations agree through p_3 and that the method of False Position requires an additional iteration to obtain the same accuracy as the Secant method. ■

	False Position	Secant	Newton
n	p_n	p_n	p_n
0	0.5	0.5	0.7853981635
1	0.7853981635	0.7853981635	0.7395361337
2	0.7363841388	0.7363841388	0.7390851781
3	0.7390581392	0.7390581392	0.7390851332
4	0.7390848638	0.7390851493	0.7390851332
5	0.7390851305	0.7390851332	
6	0.7390851332		

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Example:

Consider finding the root of $f(x) = x^2 - 3$. Let $\varepsilon = 0.01$, and start with the interval $[1, 2]$.

Example:

Consider finding the root of f(x) = x^2 - 3. Let ε = 0.01, and start with the interval [1, 2].

a	*b*	f(*a*)	f(*b*)	*c*	f(*c*)	Update	Step Size
1.0	2.0	-2.00	1.00	1.6667	-0.2221	a = c	0.6667
1.6667	2.0	-0.2221	1.0	1.7273	-0.0164	a = c	0.0606
1.7273	2.0	-0.0164	1.0	1.7317	0.0012	a = c	0.0044

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Solve:

$x^3 - 2x - 5 = 0$, by using regula falsi method ($\varepsilon = 0.0001$)

Root = 2.094547

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Sol:

Step	x0	x1	x2	f(x2)
1	2.000000	3.000000	2.058824	-0.390800
2	2.058824	3.000000	2.081264	-0.147204
3	2.081264	3.000000	2.089639	-0.054677
4	2.089639	3.000000	2.092740	-0.020203
5	2.092740	3.000000	2.093884	-0.007451
6	2.093884	3.000000	2.094305	-0.002746
7	2.094305	3.000000	2.094461	-0.001012
8	2.094461	3.000000	2.094518	-0.000373
9	2.094518	3.000000	2.094539	-0.000137
10	2.094539	3.000000	2.094547	-0.000051

Do Q 1-10 from Ex # 2.3

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Lecture # 10

Interpolation

(Interpolation & a Lagrange Polynomial)

National University of computer and emerging sciences

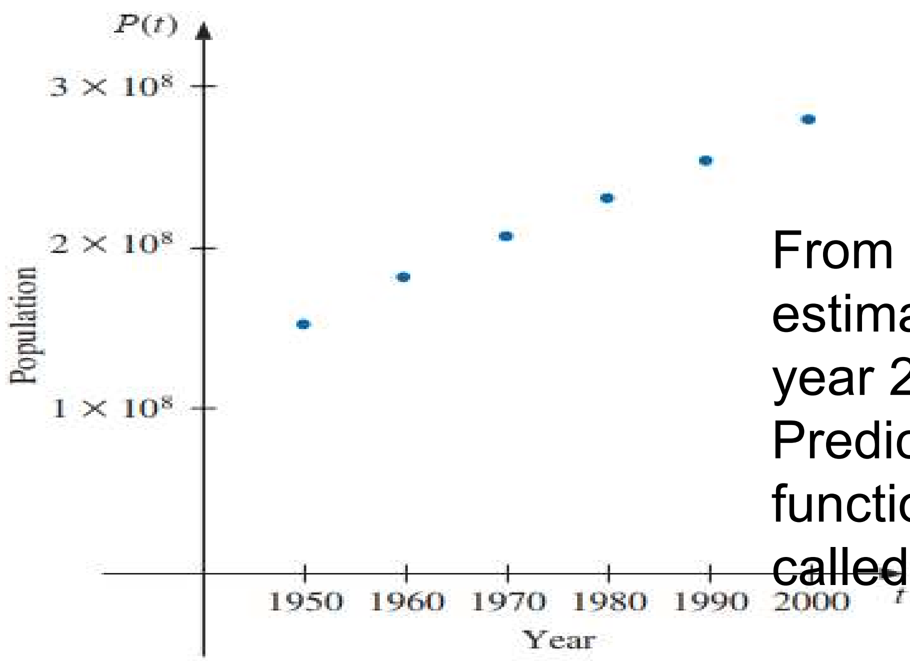
Foundation for Advancement of Science and Technology

Interpolation:

Year	1950	1960	1970	1980	1990	2000
Population (in thousands)	151,326	179,323	203,302	226,542	249,633	281,422

Interpolation:

Year	1950	1960	1970	1980	1990	2000
Population (in thousands)	151,326	179,323	203,302	226,542	249,633	281,422



From the above data (graph) some might ask to estimate population, say, in 1975 or even in the year 2020.
Predictions of this type can be obtained by using a function that fits the given data. This process is called ***interpolation.***

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Lagrange Interpolating Polynomial (1st Degree):

The linear **Lagrange interpolating polynomial** through (x_0, y_0) and (x_1, y_1) is

$$P(x) = L_0(x)f(x_0) + L_1(x)f(x_1) = \frac{x - x_1}{x_0 - x_1}f(x_0) + \frac{x - x_0}{x_1 - x_0}f(x_1).$$

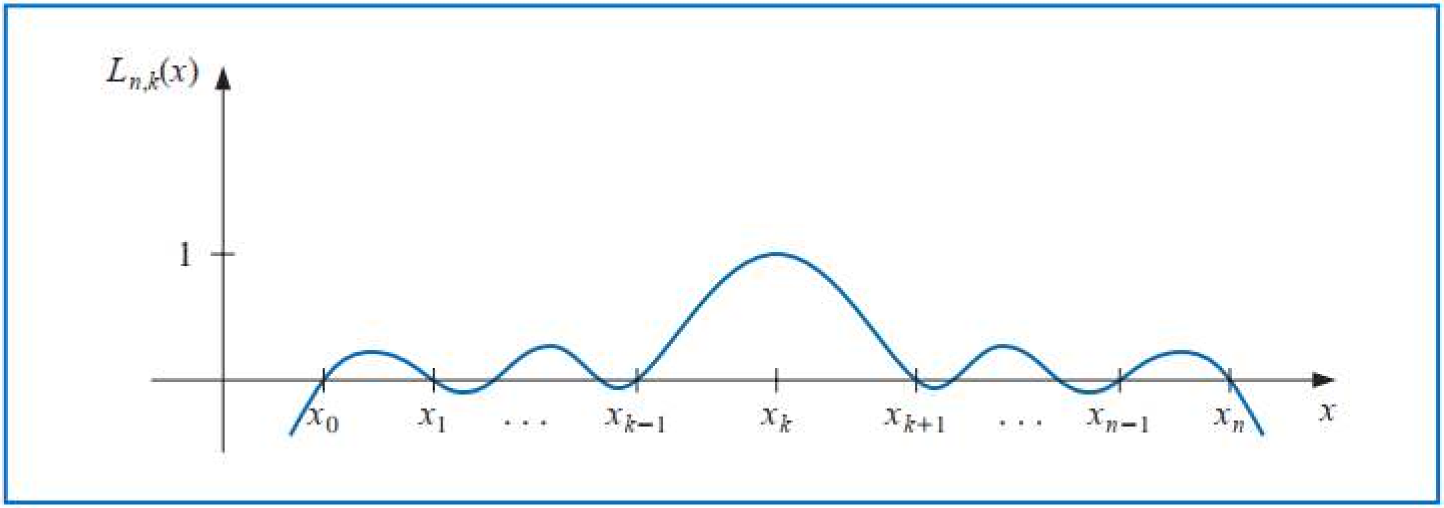
Example:

Example 1 Determine the linear Lagrange interpolating polynomial that passes through the points $(2, 4)$ and $(5, 1)$.

$$P(x) = -\frac{1}{3}(x-5)\cdot 4 + \frac{1}{3}(x-2)\cdot 1 = -\frac{4}{3}x + \frac{20}{3} + \frac{1}{3}x - \frac{2}{3} = -x + 6.$$

To generalize the concept of linear interpolation, consider the construction of a polynomial of degree at most n that passes through the $n+1$ points

Figure 3.5



The interpolating polynomial is easily described once the form of $L_{n,k}$ is known. This polynomial, called the ***n*th Lagrange interpolating polynomial**, is defined in the following theorem.

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

$$P(x) = f(x_0)L_{n,0}(x) + \cdots + f(x_n)L_{n,n}(x) = \sum_{k=0}^{n} f(x_k)L_{n,k}(x),$$

where, for each $k = 0, 1, \ldots, n$,

$$L_{n,k}(x) = \frac{(x - x_0)(x - x_1)\cdots(x - x_{k-1})(x - x_{k+1})\cdots(x - x_n)}{(x_k - x_0)(x_k - x_1)\cdots(x_k - x_{k-1})(x_k - x_{k+1})\cdots(x_k - x_n)}$$

$$= \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{(x - x_i)}{(x_k - x_i)}.$$

Theorem 3.2:

If $x_0, x_1, \ldots, x_n$ are $n+1$ distinct numbers and f is a function whose values are given at these numbers, then a unique polynomial $P(x)$ of degree at most n exists with

$$f(x_k) = P(x_k), \quad \text{for each } k = 0, 1, \ldots, n.$$

This polynomial is given by

$$P(x) = f(x_0)L_{n,0}(x) + \cdots + f(x_n)L_{n,n}(x) = \sum_{k=0}^{n} f(x_k)L_{n,k}(x), \tag{3.1}$$

National University
of computer and emerging sciences

Foundation for Advancement of Science and Technology

The symbol $\prod$ is used to write products compactly and parallels the symbol $\sum$, which is used for writing sums.

Example 2 **(a)** Use the numbers (called *nodes*) $x_0 = 2$, $x_1 = 2.75$, and $x_2 = 4$ to find the second Lagrange interpolating polynomial for $f(x) = 1/x$.

(b) Use this polynomial to approximate $f(3) = 1/3$.

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

$$L_0(x) = \frac{(x-2.75)(x-4)}{(2-2.5)(2-4)} = \frac{2}{3}(x-2.75)(x-4),$$

$$L_1(x) = \frac{(x-2)(x-4)}{(2.75-2)(2.75-4)} = -\frac{16}{15}(x-2)(x-4),$$

$$L_2(x) = \frac{(x-2)(x-2.75)}{(4-2)(4-2.5)} = \frac{2}{5}(x-2)(x-2.75).$$

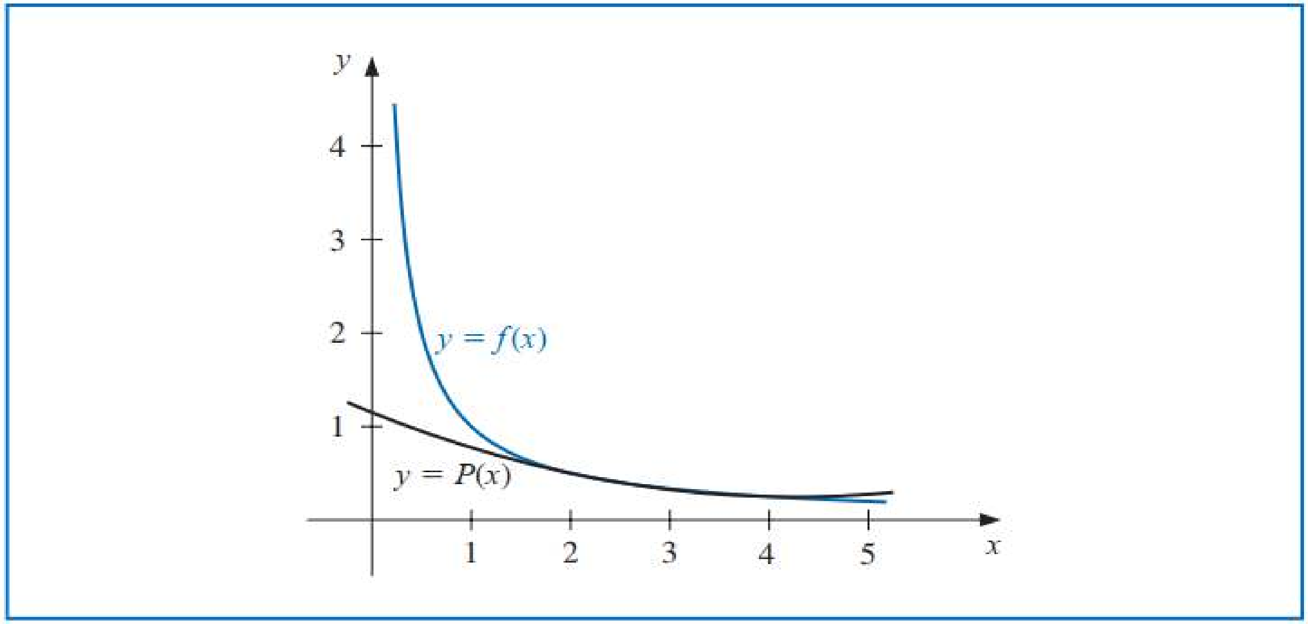
$$= \frac{1}{3}(x-2.75)(x-4) - \frac{64}{165}(x-2)(x-4) + \frac{1}{10}(x-2)(x-2.75)$$

$$= \frac{1}{22}x^2 - \frac{35}{88}x + \frac{49}{44}.$$

(b) An approximation to $f(3) = 1/3$ (see Figure 3.6) is

$$f(3) \approx P(3) = \frac{9}{22} - \frac{105}{88} + \frac{49}{44} = \frac{29}{88} \approx 0.32955.$$

Figure 3.6



National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

The algorithm of the Lagrange's interpolation

$$P(x) = \sum_{i=0}^{n} \left(\prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{(x - x_j)}{(x_i - x_j)} \right) y_i$$

Do Ex # 3.1: 1,2,5,6,13,14,19

```python
x = [0, 20, 40,50,60, 80, 100]

y = [26.0, 48.6, 61.6,66.9, 71.2, 74.8, 75.2]

m = len(x)

n = m-1 #degree of a polynomial

xp = float(input('Enter xp='))

yp = 0 # yp is summation and initially its 0

for i in range(n+1):

    p = 1

    for j in range(n+1):

        if j != i:

            p*= (xp- x[j])/(x[i]-x[j])

    yp += y[i]*p

print("For x = %.2f, %f" %(xp,yp))
```

```
Enter xp=30
For x = 30.00, 55.606563
```

National University
of computer and emerging sciences

Foundation for Advancement of Science and Technology

Lecture # 12

Divided Differences

Why Divided Difference Interpolation?

Is there any drawback of Lagrange?

Divided Difference Interpolation:

$$P_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)\cdots(x - x_{n-1})$$

Divided Difference Interpolation:

x	$f(x)$	First divided differences	Second divided differences	Third divided differences
x_0	$f[x_0]$			
		$f[x_0,x_1]=\dfrac{f[x_1]-f[x_0]}{x_1-x_0}$		
x_1	$f[x_1]$		$f[x_0,x_1,x_2]=\dfrac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0}$	
		$f[x_1,x_2]=\dfrac{f[x_2]-f[x_1]}{x_2-x_1}$		$f[x_0,x_1,x_2,x_3]=\dfrac{f[x_1,x_2,x_3]-f[x_0,x_1,x_2]}{x_3-x_0}$
x_2	$f[x_2]$		$f[x_1,x_2,x_3]=\dfrac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1}$	
		$f[x_2,x_3]=\dfrac{f[x_3]-f[x_2]}{x_3-x_2}$		$f[x_1,x_2,x_3,x_4]=\dfrac{f[x_2,x_3,x_4]-f[x_1,x_2,x_3]}{x_4-x_1}$
x_3	$f[x_3]$		$f[x_2,x_3,x_4]=\dfrac{f[x_3,x_4]-f[x_2,x_3]}{x_4-x_2}$	
		$f[x_3,x_4]=\dfrac{f[x_4]-f[x_3]}{x_4-x_3}$		$f[x_2,x_3,x_4,x_5]=\dfrac{f[x_3,x_4,x_5]-f[x_2,x_3,x_4]}{x_5-x_2}$
x_4	$f[x_4]$		$f[x_3,x_4,x_5]=\dfrac{f[x_4,x_5]-f[x_3,x_4]}{x_5-x_3}$	
		$f[x_4,x_5]=\dfrac{f[x_5]-f[x_4]}{x_5-x_4}$		
x_5	$f[x_5]$			

Algo:

Newton's Divided-Difference Formula

To obtain the divided-difference coefficients of the interpolatory polynomial P on the $(n+1)$ distinct numbers $x_0, x_1, \ldots, x_n$ for the function f:

INPUT numbers $x_0, x_1, \ldots, x_n$; values $f(x_0), f(x_1), \ldots, f(x_n)$ as $F_{0,0}, F_{1,0}, \ldots, F_{n,0}$.

OUTPUT the numbers $F_{0,0}, F_{1,1}, \ldots, F_{n,n}$ where

$$P_n(x) = F_{0,0} + \sum_{i=1}^{n} F_{i,i} \prod_{j=0}^{i-1} (x - x_j). \quad (F_{i,i} \text{ is } f[x_0, x_1, \ldots, x_i].)$$

Step 1 For $i = 1, 2, \ldots, n$
For $j = 1, 2, \ldots, i$
set $F_{i,j} = \dfrac{F_{i,j-1} - F_{i-1,j-1}}{x_i - x_{i-j}}$. $\quad (F_{i,j} = f[x_{i-j}, \ldots, x_i].)$

Step 2 OUTPUT $(F_{0,0}, F_{1,1}, \ldots, F_{n,n})$;
STOP.

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Example:

Table 3.10

x	$f(x)$
1.0	0.7651977
1.3	0.6200860
1.6	0.4554022
1.9	0.2818186
2.2	0.1103623

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Table 3.11

i	x_i	$f[x_i]$	$f[x_{i-1}, x_i]$	$f[x_{i-2}, x_{i-1}, x_i]$	$f[x_{i-3}, \ldots, x_i]$	$f[x_{i-4}, \ldots, x_i]$
0	1.0	0.7651977				
			−0.4837057			
1	1.3	0.6200860		−0.1087339		
			−0.5489460		0.0658784	
2	1.6	0.4554022		−0.0494433		0.0018251
			−0.5786120		0.0680685	
3	1.9	0.2818186		0.0118183		
			−0.5715210			
4	2.2	0.1103623				

Find $P_4(1.5)$:

$$P_4(x) = 0.7651977 - 0.4837057(x - 1.0) - 0.1087339(x - 1.0)(x - 1.3)$$
$$+ 0.0658784(x - 1.0)(x - 1.3)(x - 1.6)$$
$$+ 0.0018251(x - 1.0)(x - 1.3)(x - 1.6)(x - 1.9).$$

$$P_4(1.5) = 0.5118200$$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Lecture # 13 & 14

Forward & Backward Divided Differences

Forward Difference

Newton's divided-difference formula can be expressed in a simplified form when the nodes are arranged consecutively with equal spacing

$$P_n(x) = P_n(x_0 + sh) = f[x_0] + shf[x_0, x_1] + s(s-1)h^2 f[x_0, x_1, x_2]$$

$$+ \cdots + s(s-1)\cdots(s-n+1)h^n f[x_0, x_1, \ldots, x_n]$$

$$= f[x_0] + \sum_{k=1}^{n} s(s-1)\cdots(s-k+1)h^k f[x_0, x_1, \ldots, x_k].$$

Newton Forward-Difference Formula

$$P_n(x) = f(x_0) + \sum_{k=1}^{n} \binom{s}{k} \Delta^k f(x_0)$$

Backward Difference:

If, in addition, the nodes are equally spaced with $x = x_n + sh$ and $x = x_i + (s + n - i)h$, then

$$
\begin{aligned}
P_n(x) &= P_n(x_n + sh) \\
&= f[x_n] + sh f[x_n, x_{n-1}] + s(s+1)h^2 f[x_n, x_{n-1}, x_{n-2}] + \cdots \\
&\quad + s(s+1)\cdots(s+n-1)h^n f[x_n, \ldots, x_0].
\end{aligned}
$$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Newton Backward–Difference Formula

$$P_n(x) = f[x_n] + \sum_{k=1}^{n} (-1)^k \binom{-s}{k} \nabla^k f(x_n)$$

Previous Class Example:

The divided-difference Table 3.12 corresponds to the data in Example 1.

		First divided differences	Second divided differences	Third divided differences	Fourth divided differences
1.0	0.7651977				
		−0.4837057			
1.3	0.6200860		−0.1087339		
		−0.5489460		0.0658784	
1.6	0.4554022		−0.0494433		0.0018251
		−0.5786120		0.0680685	
1.9	0.2818186		0.0118183		
		−0.5715210			
2.2	0.1103623				

$$
\begin{aligned}
P_4(1.1) &= P_4\left(1.0 + \frac{1}{3}(0.3)\right) \\
&= 0.7651977 + \frac{1}{3}(0.3)(-0.4837057) + \frac{1}{3}\left(-\frac{2}{3}\right)(0.3)^2(-0.1087339) \\
&\quad + \frac{1}{3}\left(-\frac{2}{3}\right)\left(-\frac{5}{3}\right)(0.3)^3(0.0658784) \\
&\quad + \frac{1}{3}\left(-\frac{2}{3}\right)\left(-\frac{5}{3}\right)\left(-\frac{8}{3}\right)(0.3)^4(0.0018251) \\
&= 0.7196460.
\end{aligned}
$$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Backward:

$$P_4(2.0) = P_4\left(2.2 - \frac{2}{3}(0.3)\right)$$

$$= 0.1103623 - \frac{2}{3}(0.3)(-0.5715210) - \frac{2}{3}\left(\frac{1}{3}\right)(0.3)^2(0.0118183)$$

$$- \frac{2}{3}\left(\frac{1}{3}\right)\left(\frac{4}{3}\right)(0.3)^3(0.0680685) - \frac{2}{3}\left(\frac{1}{3}\right)\left(\frac{4}{3}\right)\left(\frac{7}{3}\right)(0.3)^4(0.0018251)$$

$$= 0.2238754. \qquad \square$$

Lecture # 17 & 18

Centered Differences (Stirling's Formula)

James Stirling (1692–1770) published this and numerous other formulas in *Methodus Differentialis* in 1720. Techniques for accelerating the convergence of various series are included in this work.

For the centered-difference formulas, we choose x_0 near the point being approximated and label the nodes directly below x_0 as $x_1, x_2, \ldots$ and those directly above as $x_{-1}, x_{-2}, \ldots$. With this convention, **Stirling's formula** is given by

$$P_n(x) = P_{2m+1}(x) = f[x_0] + \frac{sh}{2}(f[x_{-1}, x_0] + f[x_0, x_1]) + s^2h^2 f[x_{-1}, x_0, x_1] \tag{3.14}$$

$$+ \frac{s(s^2-1)h^3}{2} f[x_{-2}, x_{-1}, x_0, x_1] + f[x_{-1}, x_0, x_1, x_2])$$

$$+ \cdots + s^2(s^2-1)(s^2-4)\cdots(s^2-(m-1)^2)h^{2m} f[x_{-m}, \ldots, x_m]$$

$$+ \frac{s(s^2-1)\cdots(s^2-m^2)h^{2m+1}}{2}(f[x_{-m-1}, \ldots, x_m] + f[x_{-m}, \ldots, x_{m+1}]),$$

if $n = 2m+1$ is odd. If $n = 2m$ is even, we use the same formula but delete the last line.

x	$f(x)$	First divided differences	Second divided differences	Third divided differences	Fourth divided differences
x_{-2}	$f[x_{-2}]$				
		$f[x_{-2}, x_{-1}]$			
x_{-1}	$f[x_{-1}]$		$f[x_{-2}, x_{-1}, x_0]$		
		$f[x_{-1}, x_0]$		$f[x_{-2}, x_{-1}, x_0, x_1]$	
x_0	$f[x_0]$		$f[x_{-1}, x_0, x_1]$		$f[x_{-2}, x_{-1}, x_0, x_1, x_2]$
		$f[x_0, x_1]$		$f[x_{-1}, x_0, x_1, x_2]$	
x_1	$f[x_1]$		$f[x_0, x_1, x_2]$		
		$f[x_1, x_2]$			
x_2	$f[x_2]$				

$$P_n(x) = P_{2m+1}(x) = f[x_0] + \frac{sh}{2}(f[x_{-1}, x_0] + f[x_0, x_1]) + s^2h^2 f[x_{-1}, x_0, x_1] \qquad (3.14)$$

$$+ \frac{s(s^2-1)h^3}{2} f[x_{-2}, x_{-1}, x_0, x_1] + f[x_{-1}, x_0, x_1, x_2]) + \_\,\_\;\_\;\_$$

Example 2 Consider the table of data given in the previous examples. Use Stirling's formula to approximate $f(1.5)$ with $x_0 = 1.6$.

x	$f(x)$	First divided differences	Second divided differences	Third divided differences	Fourth divided differences
1.0	0.7651977				
		−0.4837057			
1.3	0.6200860		−0.1087339		
		−0.5489460		0.0658784	
1.6	0.4554022		−0.0494433		0.0018251
		−0.5786120		0.0680685	
1.9	0.2818186		0.0118183		
		−0.5715210			
2.2	0.1103623				

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

The formula, with $h = 0.3$, $x_0 = 1.6$, and $s = -\frac{1}{3}$, becomes

$$f(1.5) \approx P_4\left(1.6 + \left(-\frac{1}{3}\right)(0.3)\right)$$

$$= 0.4554022 + \left(-\frac{1}{3}\right)\left(\frac{0.3}{2}\right)((-0.5489460) + (-0.5786120))$$

$$+ \left(-\frac{1}{3}\right)^2 (0.3)^2(-0.0494433)$$

$$+ \frac{1}{2}\left(-\frac{1}{3}\right)\left(\left(-\frac{1}{3}\right)^2 - 1\right)(0.3)^3(0.0658784 + 0.0680685)$$

$$+ \left(-\frac{1}{3}\right)^2\left(\left(-\frac{1}{3}\right)^2 - 1\right)(0.3)^4(0.0018251) = 0.5118200.$$

Do Q 1,2,7,8 from Ex # 3.3

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Lecture # 19

Numerical Differentiation

Recall!

The derivative of the function f at x_0 is

$$f'(x_0) = \lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}.$$

This formula gives an obvious way to generate an approximation to $f'(x_0)$; simply compute

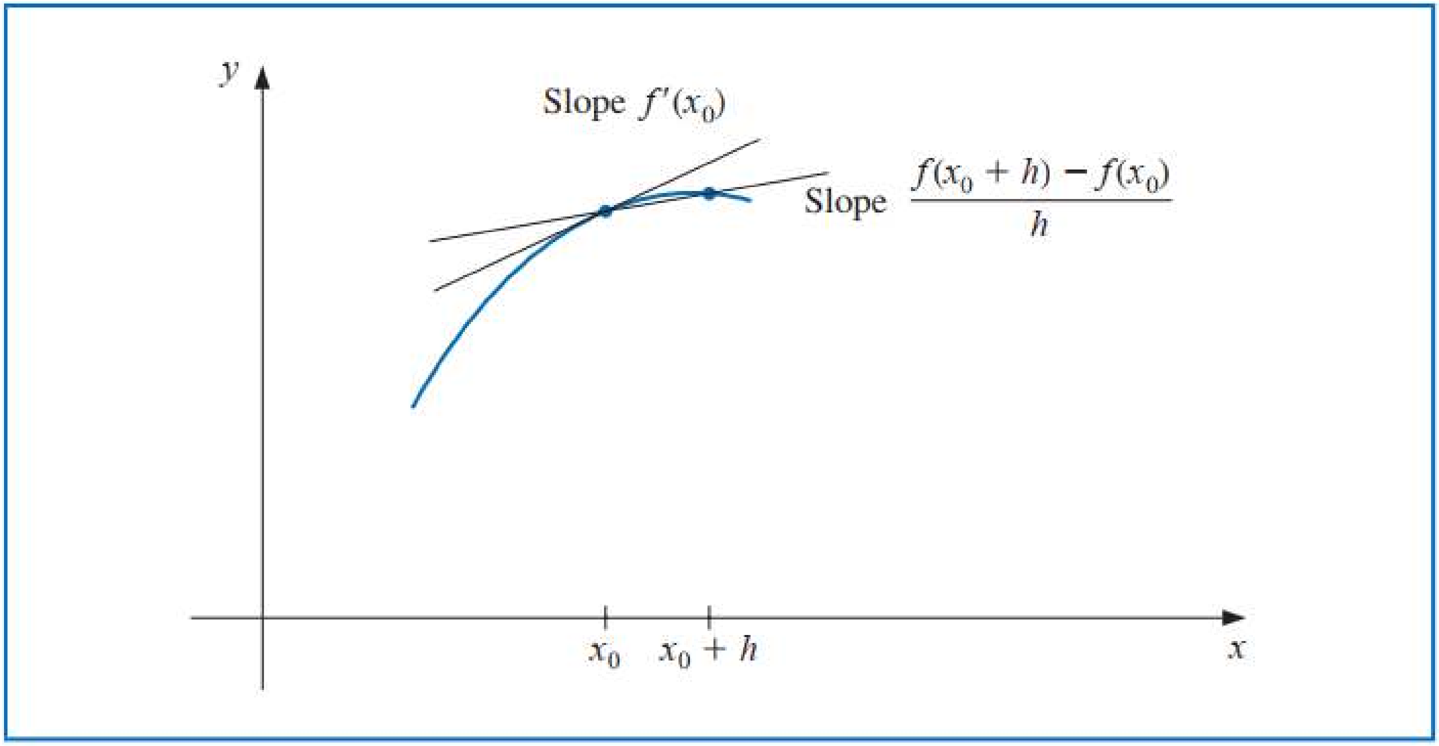
$$\frac{f(x_0 + h) - f(x_0)}{h}$$

for small values of h. Although this may be obvious, it is not very successful, due to our old nemesis round-off error. But it is certainly a place to start.

$$f'(x_0) = \frac{f(x_0+h) - f(x_0)}{h} - \frac{h}{2} f''(\xi).$$

Error

Figure 4.1



Example 1 Use the forward-difference formula to approximate the derivative of $f(x) = \ln x$ at $x_0 = 1.8$ using $h = 0.1$, $h = 0.05$, and $h = 0.01$, and determine bounds for the approximation errors.

Solution The forward-difference formula

$$\frac{f(1.8+h)-f(1.8)}{h}$$

Solution The forward-difference formula

$$\frac{f(1.8+h) - f(1.8)}{h}$$

with $h = 0.1$ gives

$$\frac{\ln 1.9 - \ln 1.8}{0.1} = \frac{0.64185389 - 0.58778667}{0.1} = 0.5406722.$$

Because $f''(x) = -1/x^2$ and $1.8 < \xi < 1.9$, a bound for this approximation error is

$$\frac{|hf''(\xi)|}{2} = \frac{|h|}{2\xi^2} < \frac{0.1}{2(1.8)^2} = 0.0154321.$$

Table 4.1

h	$f(1.8+h)$	$\dfrac{f(1.8+h)-f(1.8)}{h}$	$\dfrac{\lvert h\rvert}{2(1.8)^2}$
0.1	0.64185389	0.5406722	0.0154321
0.05	0.61518564	0.5479795	0.0077160
0.01	0.59332685	0.5540180	0.0015432

Since $f'(x) = 1/x$, the exact value of $f'(1.8)$ is $0.55\overline{5}$, and in this case the error bounds are quite close to the true approximation error. ■

Three-Point Formulas

$$f'(x_0) = \frac{1}{2h}[-3f(x_0) + 4f(x_0 + h) - f(x_0 + 2h)] + \frac{h^2}{3} f^{(3)}(\xi_0),$$

$$f'(x_0) = \frac{1}{2h}[-f(x_0 - h) + f(x_0 + h)] - \frac{h^2}{6} f^{(3)}(\xi_1),$$

and

$$f'(x_0) = \frac{1}{2h}[f(x_0 - 2h) - 4f(x_0 - h) + 3f(x_0)] + \frac{h^2}{3} f^{(3)}(\xi_2).$$

Finally, note that the last of these equations can be obtained from the first by simply replacing h with $-h$, so there are actually only two formulas:

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Three-Point Endpoint Formula

- $$f'(x_0) = \frac{1}{2h}[-3f(x_0) + 4f(x_0 + h) - f(x_0 + 2h)] + \frac{h^2}{3}f^{(3)}(\xi_0), \quad (4.4)$$

where ξ_0 lies between x_0 and $x_0 + 2h$.

Three-Point Midpoint Formula

- $$f'(x_0) = \frac{1}{2h}[f(x_0 + h) - f(x_0 - h)] - \frac{h^2}{6} f^{(3)}(\xi_1),$$

where ξ_1 lies between $x_0 - h$ and $x_0 + h$.

Five-Point Midpoint Formula

- $$f'(x_0) = \frac{1}{12h}[f(x_0 - 2h) - 8f(x_0 - h) + 8f(x_0 + h) - f(x_0 + 2h)] + \frac{h^4}{30} f^{(5)}(\xi), \tag{4.6}$$

where ξ lies between $x_0 - 2h$ and $x_0 + 2h$.

Five-Point Endpoint Formula

- $$f'(x_0) = \frac{1}{12h}[-25f(x_0) + 48f(x_0+h) - 36f(x_0+2h) + 16f(x_0+3h) - 3f(x_0+4h)] + \frac{h^4}{5}f^{(5)}(\xi), \tag{4.7}$$

where ξ lies between x_0 and $x_0 + 4h$.

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Lecture # 20-21

Numerical Differentiation

Example 2 Values for $f(x) = xe^x$ are given in Table 4.2. Use all the applicable three-point and five-point formulas to approximate $f'(2.0)$.

Table 4.2

x	$f(x)$
1.8	10.889365
1.9	12.703199
2.0	14.778112
2.1	17.148957
2.2	19.855030

Three-Point Endpoint Formula

- $$f'(x_0) = \frac{1}{2h}[-3f(x_0) + 4f(x_0+h) - f(x_0+2h)] + \frac{h^2}{3}f^{(3)}(\xi_0), \tag{4.4}$$

where ξ_0 lies between x_0 and $x_0 + 2h$.

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Three-Point Endpoint Formula

- $$f'(x_0) = \frac{1}{2h}[-3f(x_0) + 4f(x_0 + h) - f(x_0 + 2h)] + \frac{h^2}{3} f^{(3)}(\xi_0), \tag{4.4}$$

where ξ_0 lies between x_0 and $x_0 + 2h$.

Using the endpoint formula (4.4) with $h = 0.1$ gives

$$\frac{1}{0.2}[-3f(2.0) + 4f(2.1) - f(2.2] = 5[-3(14.778112) + 4(17.148957) - 19.855030)] = 22.032310,$$

$$f'(x_0) = \frac{1}{2h}[f(x_0 - 2h) - 4f(x_0 - h) + 3f(x_0)] + \frac{h^2}{3} f^{(3)}(\xi_2).$$

and with $h = -0.1$ gives 22.054525.

Three-Point Midpoint Formula

- $$f'(x_0) = \frac{1}{2h}[f(x_0 + h) - f(x_0 - h)] - \frac{h^2}{6} f^{(3)}(\xi_1), \tag{4.5}$$

Three-Point Midpoint Formula

- $$f'(x_0) = \frac{1}{2h}[f(x_0+h) - f(x_0-h)] - \frac{h^2}{6} f^{(3)}(\xi_1), \tag{4.5}$$

Using the midpoint formula (4.5) with $h = 0.1$ gives

$$\frac{1}{0.2}[f(2.1) - f(1.9)] = 5(17.148957 - 12.7703199) = 22.228790,$$

and with $h = 0.2$ gives 22.414163.

Five-Point Midpoint Formula

- $$f'(x_0) = \frac{1}{12h}[f(x_0 - 2h) - 8f(x_0 - h) + 8f(x_0 + h) - f(x_0 + 2h)] + \frac{h^4}{30} f^{(5)}(\xi), \tag{4.6}$$

where ξ lies between $x_0 - 2h$ and $x_0 + 2h$.

x	$f(x)$
1.8	10.889365
1.9	12.703199
2.0	14.778112
2.1	17.148957
2.2	19.855030

Five-Point Endpoint Formula

- $$f'(x_0) = \frac{1}{12h}[-25f(x_0) + 48f(x_0+h) - 36f(x_0+2h) + 16f(x_0+3h) - 3f(x_0+4h)] + \frac{h^4}{5}f^{(5)}(\xi), \quad (4.7)$$

where ξ lies between x_0 and $x_0 + 4h$.

x	$f(x)$
1.8	10.889365
1.9	12.703199
2.0	14.778112
2.1	17.148957
2.2	19.855030

The only five-point formula for which the table gives sufficient data is the midpoint formula (4.6) with $h = 0.1$. This gives

$$\frac{1}{1.2}[f(1.8) - 8f(1.9) + 8f(2.1) - f(2.2)] = \frac{1}{1.2}[10.889365 - 8(12.703199) + 8(17.148957) - 19.855030] = 22.166999$$

If we had no other information we would accept the five-point midpoint approximation using $h = 0.1$ as the most accurate, and expect the true value to be between that approximation and the three-point mid-point approximation that is in the interval [22.166, 22.229].

The true value in this case is $f'(2.0) = (2+1)e^2 = 22.167168$, so the approximation errors are actually:

Three-point endpoint with $h = 0.1$: 1.35×10^{-1};

Three-point endpoint with $h = -0.1$: 1.13×10^{-1};

Three-point midpoint with $h = 0.1$: -6.16×10^{-2};

Three-point midpoint with $h = 0.2$: -2.47×10^{-1};

Five-point midpoint with $h = 0.1$: 1.69×10^{-4}. ■

Second Derivative Midpoint Formula

- $$f''(x_0) = \frac{1}{h^2}[f(x_0 - h) - 2f(x_0) + f(x_0 + h)] - \frac{h^2}{12} f^{(4)}(\xi), \qquad (4.9)$$

for some ξ, where $x_0 - h < \xi < x_0 + h$.

In Example 2 we used the data shown in Table 4.3 to approximate the first derivative of $f(x) = xe^x$ at $x = 2.0$. Use the second derivative formula (4.9) to approximate $f''(2.0)$.

Table 4.3

x	$f(x)$
1.8	10.889365
1.9	12.703199
2.0	14.778112
2.1	17.148957
2.2	19.855030

Solution The data permits us to determine two approximations for $f''(2.0)$. Using (4.9) with $h = 0.1$ gives

$$\frac{1}{0.01}[f(1.9) - 2f(2.0) + f(2.1)] = 100[12.703199 - 2(14.778112) + 17.148957]$$
$$= 29.593200,$$

and using (4.9) with $h = 0.2$ gives

$$\frac{1}{0.04}[f(1.8) - 2f(2.0) + f(2.2)] = 25[10.889365 - 2(14.778112) + 19.855030]$$
$$= 29.704275.$$

Because $f''(x) = (x + 2)e^x$, the exact value is $f''(2.0) = 29.556224$. Hence the actual errors are -3.70×10^{-2} and -1.48×10^{-1}, respectively. ■

Problems related to Numerical Differentiation (Ex # 4.1)

1. Use the forward-difference formulas and backward-difference formulas to determine each missing entry in the following tables.

a.

x	$f(x)$	$f'(x)$
0.5	0.4794	
0.6	0.5646	
0.7	0.6442	

b.

x	$f(x)$	$f'(x)$
0.0	0.00000	
0.2	0.74140	
0.4	1.3718	

2. Use the forward-difference formulas and backward-difference formulas to determine each missing entry in the following tables.

a.

x	$f(x)$	$f'(x)$
−0.3	1.9507	
−0.2	2.0421	
−0.1	2.0601	

b.

x	$f(x)$	$f'(x)$
1.0	1.0000	
1.2	1.2625	
1.4	1.6595	

5. Use the most accurate three-point formula to determine each missing entry in the following tables.

a.

x	$f(x)$	$f'(x)$
1.1	9.025013	
1.2	11.02318	
1.3	13.46374	
1.4	16.44465	

b.

x	$f(x)$	$f'(x)$
8.1	16.94410	
8.3	17.56492	
8.5	18.19056	
8.7	18.82091	

c.

x	$f(x)$	$f'(x)$
2.9	−4.827866	
3.0	−4.240058	
3.1	−3.496909	
3.2	−2.596792	

d.

x	$f(x)$	$f'(x)$
2.0	3.6887983	
2.1	3.6905701	
2.2	3.6688192	
2.3	3.6245909	

6. Use the most accurate three-point formula to determine each missing entry in the following tables.

a.

x	$f(x)$	$f'(x)$
-0.3	-0.27652	
-0.2	-0.25074	
-0.1	-0.16134	
0	0	

b.

x	$f(x)$	$f'(x)$
7.4	-68.3193	
7.6	-71.6982	
7.8	-75.1576	
8.0	-78.6974	

c.

x	$f(x)$	$f'(x)$
1.1	1.52918	
1.2	1.64024	
1.3	1.70470	
1.4	1.71277	

d.

x	$f(x)$	$f'(x)$
-2.7	0.054797	
-2.5	0.11342	
-2.3	0.65536	
-2.1	0.98472	

18. Consider the following table of data:

x	0.2	0.4	0.6	0.8	1.0
$f(x)$	0.9798652	0.9177710	0.808038	0.6386093	0.3843735

a. Use all the appropriate formulas given in this section to approximate $f'(0.4)$ and $f''(0.4)$.

b. Use all the appropriate formulas given in this section to approximate $f'(0.6)$ and $f''(0.6)$.

25. In Exercise 10 of Section 3.4 data were given describing a car traveling on a straight road. That problem asked to predict the position and speed of the car when $t = 10$ s. Use the following times and positions to predict the speed at each time listed.

Time	0	3	5	8	10	13
Distance	0	225	383	623	742	993

26. In a circuit with impressed voltage $\mathcal{E}(t)$ and inductance L, Kirchhoff's first law gives the relationship

$$\mathcal{E}(t) = L\frac{di}{dt} + Ri,$$

where R is the resistance in the circuit and i is the current. Suppose we measure the current for several values of t and obtain:

t	1.00	1.01	1.02	1.03	1.0
i	3.10	3.12	3.14	3.18	3.24

where t is measured in seconds, i is in amperes, the inductance L is a constant 0.98 henries, and the resistance is 0.142 ohms. Approximate the voltage $\mathcal{E}(t)$ when $t = 1.00, 1.01, 1.02, 1.03$, and 1.04.

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Lecture # 23-24

Numerical Integration

Elements of Numerical Integration

The need often arises for evaluating the definite integral of a function that has no explicit antiderivative or whose antiderivative is not easy to obtain. The basic method involved in approximating $\int_a^b f(x)\,dx$ is called **numerical quadrature**. It uses a sum $\sum_{i=0}^{n} a_i f(x_i)$ to approximate $\int_a^b f(x)\,dx$.

RECALL!

$$\boxed{P_n(x) = \sum_{i=0}^{n} f(x_i)L_i(x)}$$

and its truncation error term over $[a, b]$ to obtain

$$\int_a^b f(x)\,dx = \int_a^b \sum_{i=0}^{n} f(x_i)L_i(x)\,dx + \int_a^b \prod_{i=0}^{n}(x - x_i)\frac{f^{(n+1)}(\xi(x))}{(n+1)!}\,dx$$

$$= \sum_{i=0}^{n} a_i f(x_i) + \frac{1}{(n+1)!}\int_a^b \prod_{i=0}^{n}(x - x_i) f^{(n+1)}(\xi(x))\,dx,$$

where $\xi(x)$ is in $[a, b]$ for each x and

$$a_i = \int_a^b L_i(x)\,dx, \quad \text{for each } i = 0, 1, \ldots, n.$$

The quadrature formula is, therefore,

$$\int_a^b f(x)\,dx \approx \sum_{i=0}^{n} a_i f(x_i),$$

Before discussing the general situation of quadrature formulas, let us consider formulas produced by using first and second Lagrange polynomials with equally-spaced nodes. This gives the **Trapezoidal rule** and **Simpson's rule**, which are commonly introduced in calculus courses.

The Trapezoidal Rule

To derive the Trapezoidal rule for approximating $\int_a^b f(x)\,dx$, let $x_0 = a$, $x_1 = b$, $\boxed{h = b - a}$ and use the linear Lagrange polynomial:

$$P_1(x) = \frac{(x - x_1)}{(x_0 - x_1)} f(x_0) + \frac{(x - x_0)}{(x_1 - x_0)} f(x_1).$$

Then

$$\int_a^b f(x)\,dx = \int_{x_0}^{x_1} \left[\frac{(x - x_1)}{(x_0 - x_1)} f(x_0) + \frac{(x - x_0)}{(x_1 - x_0)} f(x_1) \right] dx + \frac{1}{2} \int_{x_0}^{x_1} f''(\xi(x))(x - x_0)(x - x_1)\,dx. \quad (4.23)$$

If you wish to completely observe its derivation, Kindly refer TEXTBOOK

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

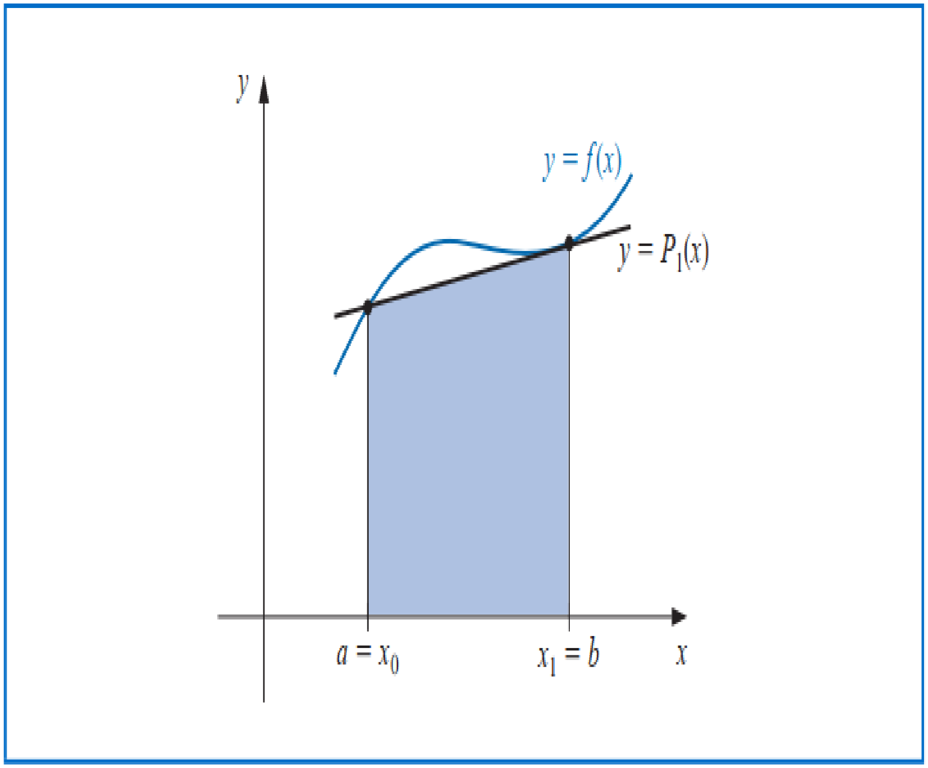
The Trapezoidal Rule

$$\int_a^b f(x)\,dx = \frac{h}{2}[f(x_0) + f(x_1)] - \frac{h^3}{12} f''(\xi).$$

This is called the Trapezoidal rule because when f is a function with positive values, $\int_a^b f(x)\,dx$ is approximated by the area in a trapezoid, as shown in Figure 4.3.

When we use the term *trapezoid* we mean a four-sided figure that has at least two of its sides parallel. The European term for this figure is *trapezium*. To further confuse the issue, the European word trapezoidal refers to a four-sided figure with no sides equal, and the American word for this type of figure is trapezium.

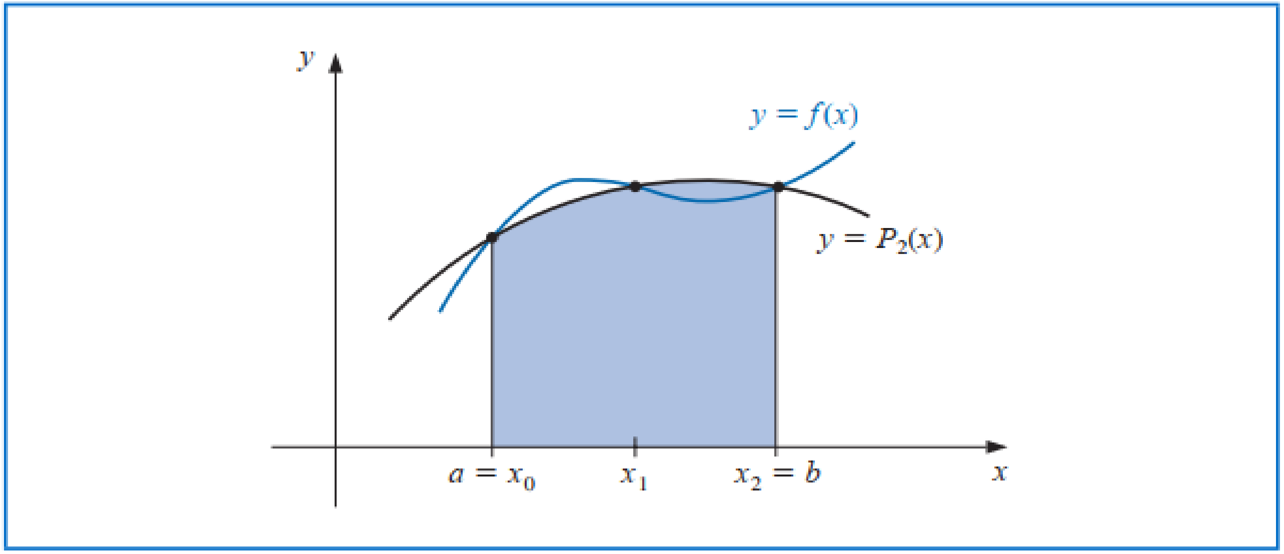
Figure 4.3



Simpson's Rule

Simpson's rule results from integrating over $[a, b]$ the second Lagrange polynomial with equally-spaced nodes $x_0 = a$, $x_2 = b$, and $x_1 = a + h$, where $\boxed{h = (b - a)/2.}$ (See Figure 4.4.)

Figure 4.4



Therefore

$$\int_a^b f(x)\,dx = \int_{x_0}^{x_2} \left[\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f(x_1) + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} f(x_2) \right] dx$$

$$+ \int_{x_0}^{x_2} \frac{(x-x_0)(x-x_1)(x-x_2)}{6} f^{(3)}(\xi(x))\,dx.$$

error

Simpson's Rule:

$$\int_{x_0}^{x_2} f(x)\,dx = \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)] - \frac{h^5}{90} f^{(4)}(\xi).$$

Thomas Simpson (1710–1761) was a self-taught mathematician who supported himself during his early years as a weaver. His primary interest was probability theory, although in 1750 he published a two-volume calculus book entitled *The Doctrine and Application of Fluxions.*

Example 1 Compare the Trapezoidal rule and Simpson's rule approximations to $\int_0^2 f(x)\,dx$ when $f(x)$ is

(a) x^2 **(b)** x^4 **(c)** $(x+1)^{-1}$

(d) $\sqrt{1+x^2}$ **(e)** $\sin x$ **(f)** e^x

T

$$\int_{x_0}^{x_1} f(x)\,dx = \frac{h}{2}[f(x_0) + f(x_1)] - \frac{h^3}{12} f''(\xi), \quad \text{where} \quad x_0 < \xi < x_1. \tag{4.25}$$

S

$$\int_{x_0}^{x_2} f(x)\,dx = \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)] - \frac{h^5}{90} f^{(4)}(\xi), \quad \text{where} \quad x_0 < \xi < x_2. \tag{4.26}$$

Solution On $[0, 2]$ the Trapezoidal and Simpson's rule have the forms

$$\text{Trapezoid:} \quad \int_0^2 f(x)\,dx \approx f(0) + f(2) \quad \text{and}$$

$$\text{Simpson's:} \quad \int_0^2 f(x)\,dx \approx \frac{1}{3}[f(0) + 4f(1) + f(2)].$$

When $f(x) = x^2$ they give

$$\text{Trapezoid:} \quad \int_0^2 f(x)\, dx \approx 0^2 + 2^2 = 4 \quad \text{and}$$

$$\text{Simpson's:} \quad \int_0^2 f(x)\, dx \approx \frac{1}{3}[(0^2) + 4 \cdot 1^2 + 2^2] = \frac{8}{3}.$$

The approximation from Simpson's rule is exact because its truncation error involves $f^{(4)}$, which is identically 0 when $f(x) = x^2$.

Table 4.7

	(a)	(b)	(c)	(d)	(e)	(f)
$f(x)$	x^2	x^4	$(x+1)^{-1}$	$\sqrt{1+x^2}$	$\sin x$	e^x
Exact value	2.667	6.400	1.099	2.958	1.416	6.389
Trapezoidal	4.000	16.000	1.333	3.326	0.909	8.389
Simpson's	2.667	6.667	1.111	2.964	1.425	6.421

The improved accuracy of Simpson's rule over the Trapezoidal rule is intuitively explained by the fact that Simpson's rule includes a midpoint evaluation that provides better balance to the approximation.

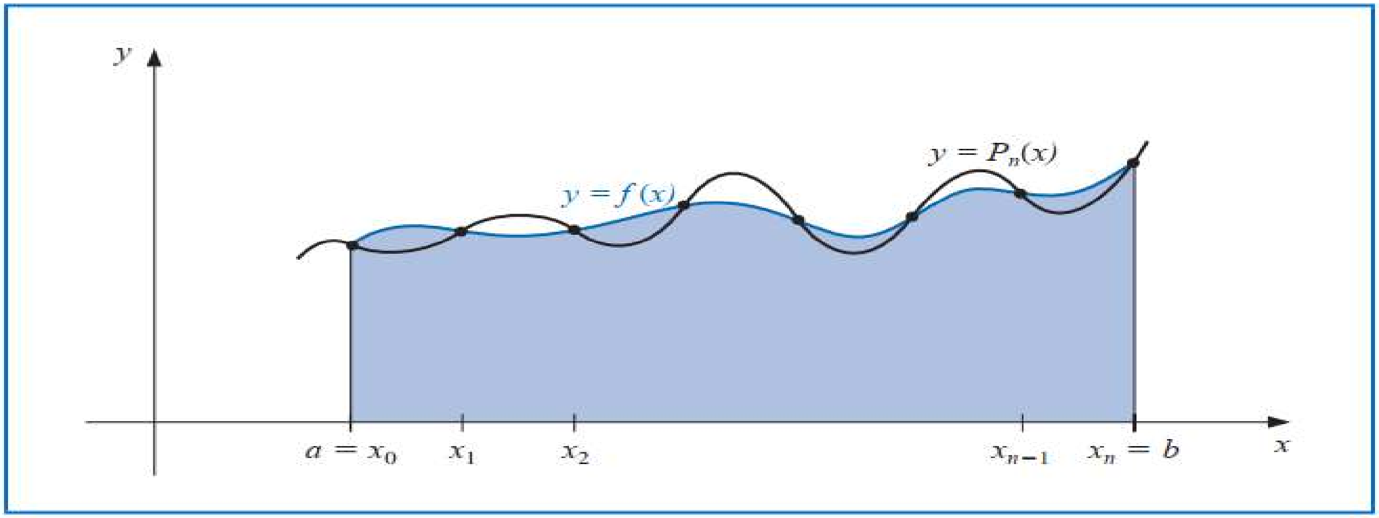
The open and closed terminology for methods implies that the open methods use as nodes only points in the open interval, (a, b) to approximate $\int_a^b f(x)\,dx$. The closed methods include the points a and b of the closed interval $[a, b]$ as nodes.

The Trapezoidal and Simpson's rules are examples of a class of methods known as Newton Cotes formulas. There are two types of Newton-Cotes formulas, open and closed.

Closed Newton-Cotes Formulas

The $(n+1)$-*point closed Newton-Cotes formula* uses nodes $x_i = x_0 + ih$, for $i = 0, 1, \ldots, n$, where $x_0 = a$, $x_n = b$ and $h = (b - a)/n$. (See Figure 4.5.) It is called closed because the endpoints of the closed interval $[a, b]$ are included as nodes.

Figure 4.5



Some of the common **closed Newton-Cotes formulas** with their error terms are listed. Note that in each case the unknown value ξ lies in (a, b).

$n = 1$: Trapezoidal rule

$$\int_{x_0}^{x_1} f(x)\,dx = \frac{h}{2}[f(x_0) + f(x_1)] - \frac{h^3}{12} f''(\xi), \quad \text{where} \quad x_0 < \xi < x_1. \tag{4.25}$$

$n = 2$: Simpson's rule

$$\int_{x_0}^{x_2} f(x)\,dx = \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)] - \frac{h^5}{90} f^{(4)}(\xi), \quad \text{where} \quad x_0 < \xi < x_2. \tag{4.26}$$

$n = 3$: Simpson's Three-Eighths rule

$$\int_{x_0}^{x_3} f(x)\,dx = \frac{3h}{8}[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)] - \frac{3h^5}{80} f^{(4)}(\xi), \qquad (4.27)$$

$$\text{where} \quad x_0 < \xi < x_3.$$

$n = 4$:

$$\int_{x_0}^{x_4} f(x)\,dx = \frac{2h}{45}[7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)] - \frac{8h^7}{945} f^{(6)}(\xi),$$

$$\text{where} \quad x_0 < \xi < x_4. \qquad (4.28)$$

Open Newton-Cotes Formulas

The *open Newton-Cotes formulas* do not include the endpoints of $[a, b]$ as nodes. They use the nodes $x_i = x_0 + ih$, for each $i = 0, 1, \ldots, n$, where $h = (b - a)/(n + 2)$ and $x_0 = a + h$. This implies that $x_n = b - h$, so we label the endpoints by setting $x_{-1} = a$ and $x_{n+1} = b$, Some of the common **open Newton-Cotes** formulas with their error terms are as follows:

$n = 0$: Midpoint rule

$$\int_{x_{-1}}^{x_1} f(x)\,dx = 2hf(x_0) + \frac{h^3}{3}f''(\xi), \quad \text{where} \quad x_{-1} < \xi < x_1. \qquad (4.29)$$

$n = 1$:

$$\int_{x_{-1}}^{x_2} f(x)\,dx = \frac{3h}{2}[f(x_0) + f(x_1)] + \frac{3h^3}{4}f''(\xi), \quad \text{where} \quad x_{-1} < \xi < x_2. \qquad (4.30)$$

Compare the results of the closed and open Newton-Cotes formulas listed as (4.25)–(4.28) and (4.29)–(4.32) when approximating

$$\int_0^{\pi/4} \sin x \, dx = 1 - \sqrt{2}/2 \approx 0.29289322.$$

Solution For the closed formulas we have

$$n=1: \quad \frac{(\pi/4)}{2}\left[\sin 0+\sin\frac{\pi}{4}\right] \approx 0.27768018$$

$$n=2: \quad \frac{(\pi/8)}{3}\left[\sin 0+4\sin\frac{\pi}{8}+\sin\frac{\pi}{4}\right] \approx 0.29293264$$

$$n=3: \quad \frac{3(\pi/12)}{8}\left[\sin 0+3\sin\frac{\pi}{12}+3\sin\frac{\pi}{6}+\sin\frac{\pi}{4}\right] \approx 0.29291070$$

$$n=4: \quad \frac{2(\pi/16)}{45}\left[7\sin 0+32\sin\frac{\pi}{16}+12\sin\frac{\pi}{8}+32\sin\frac{3\pi}{16}+7\sin\frac{\pi}{4}\right] \approx 0.29289318$$

and for the open formulas we have

$$n = 0: \quad 2(\pi/8)\left[\sin\frac{\pi}{8}\right] \approx 0.30055887$$

$$n = 1: \quad \frac{3(\pi/12)}{2}\left[\sin\frac{\pi}{12} + \sin\frac{\pi}{6}\right] \approx 0.29798754$$

$$n = 2: \quad \frac{4(\pi/16)}{3}\left[2\sin\frac{\pi}{16} - \sin\frac{\pi}{8} + 2\sin\frac{3\pi}{16}\right] \approx 0.29285866$$

$$n = 3: \quad \frac{5(\pi/20)}{24}\left[11\sin\frac{\pi}{20} + \sin\frac{\pi}{10} + \sin\frac{3\pi}{20} + 11\sin\frac{\pi}{5}\right] \approx 0.29286923$$

EXERCISE SET 4.3

1. Approximate the following integrals using the Trapezoidal rule.

   a. $\int_{0.5}^{1} x^4\,dx$
   b. $\int_{0}^{0.5} \frac{2}{x-4}\,dx$
   c. $\int_{1}^{1.5} x^2 \ln x\,dx$
   d. $\int_{0}^{1} x^2 e^{-x}\,dx$
   e. $\int_{1}^{1.6} \frac{2x}{x^2-4}\,dx$
   f. $\int_{0}^{0.35} \frac{2}{x^2-4}\,dx$
   g. $\int_{0}^{\pi/4} x \sin x\,dx$
   h. $\int_{0}^{\pi/4} e^{3x} \sin 2x\,dx$

2. Approximate the following integrals using the Trapezoidal rule.

   a. $\int_{-0.25}^{0.25} (\cos x)^2\,dx$
   b. $\int_{-0.5}^{0} x \ln(x+1)\,dx$
   c. $\int_{0.75}^{1.3} \left((\sin x)^2 - 2x \sin x + 1\right)\,dx$
   d. $\int_{e}^{e+1} \frac{1}{x \ln x}\,dx$

3. Find a bound for the error in Exercise 1 using the error formula, and compare this to the actual error.
4. Find a bound for the error in Exercise 2 using the error formula, and compare this to the actual error.
5. Repeat Exercise 1 using Simpson's rule.
6. Repeat Exercise 2 using Simpson's rule.
7. Repeat Exercise 3 using Simpson's rule and the results of Exercise 5.
8. Repeat Exercise 4 using Simpson's rule and the results of Exercise 6.
9. Repeat Exercise 1 using the Midpoint rule.
10. Repeat Exercise 2 using the Midpoint rule.

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Lecture # 25 & 26

Composite Numerical Integration

Composite Numerical Integration

The Newton-Cotes formulas are generally unsuitable for use over large integration intervals. High-degree formulas would be required, and the values of the coefficients in these formulas are difficult to obtain. Also, the Newton-Cotes formulas are based on interpolatory polynomials that use equally-spaced nodes, a procedure that is inaccurate over large intervals because of the oscillatory nature of high-degree polynomials.

In this section, we discuss a *piecewise* approach to numerical integration that uses the low-order Newton-Cotes formulas. These are the techniques most often applied.

Example 1 Use Simpson's rule to approximate $\int_0^4 e^x\,dx$ and compare this to the results obtained by adding the Simpson's rule approximations for $\int_0^2 e^x\,dx$ and $\int_2^4 e^x\,dx$. Compare these approximations to the sum of Simpson's rule for $\int_0^1 e^x\,dx$, $\int_1^2 e^x\,dx$, $\int_2^3 e^x\,dx$, and $\int_3^4 e^x\,dx$.

Solution Simpson's rule on $[0, 4]$ uses $h = 2$ and gives

$$\int_0^4 e^x\,dx \approx \frac{2}{3}(e^0 + 4e^2 + e^4) = 56.76958.$$

The exact answer in this case is $e^4 - e^0 = 53.59815$, and the error -3.17143 is far larger than we would normally accept.

Applying Simpson's rule on each of the intervals $[0, 2]$ and $[2, 4]$ uses $h = 1$ and gives

$$\begin{aligned}
\int_0^4 e^x\,dx &= \int_0^2 e^x\,dx + \int_2^4 e^x\,dx \\
&\approx \frac{1}{3}\left(e^0 + 4e + e^2\right) + \frac{1}{3}\left(e^2 + 4e^3 + e^4\right) \\
&= \frac{1}{3}\left(e^0 + 4e + 2e^2 + 4e^3 + e^4\right) \\
&= 53.86385.
\end{aligned}$$

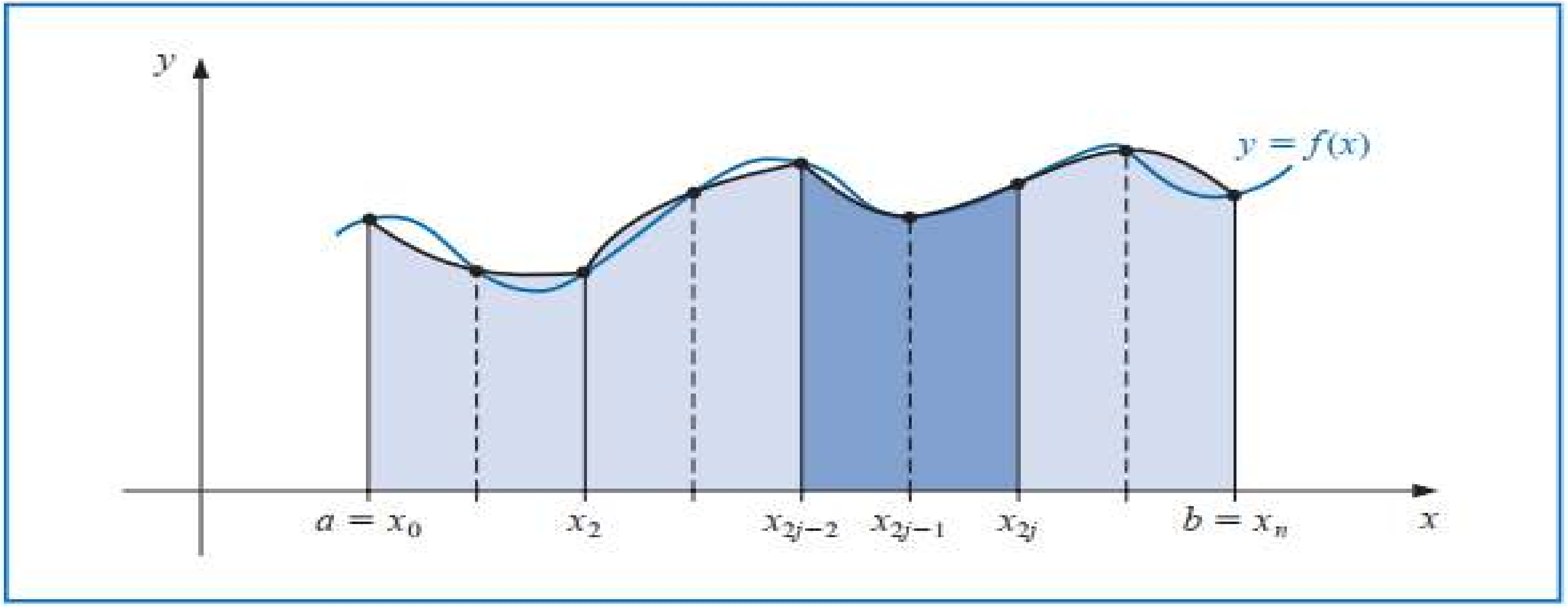
For the integrals on [0, 1],[1, 2],[3, 4], and [3, 4] we use Simpson's rule four times with $h = \frac{1}{2}$ giving

$$\begin{aligned}
\int_0^4 e^x\,dx &= \int_0^1 e^x\,dx + \int_1^2 e^x\,dx + \int_2^3 e^x\,dx + \int_3^4 e^x\,dx \\
&\approx \frac{1}{6}\left(e_0 + 4e^{1/2} + e\right) + \frac{1}{6}\left(e + 4e^{3/2} + e^2\right) \\
&\quad + \frac{1}{6}\left(e^2 + 4e^{5/2} + e^3\right) + \frac{1}{6}\left(e^3 + 4e^{7/2} + e^4\right) \\
&= \frac{1}{6}\left(e^0 + 4e^{1/2} + 2e + 4e^{3/2} + 2e^2 + 4e^{5/2} + 2e^3 + 4e^{7/2} + e^4\right) \\
&= 53.61622.
\end{aligned}$$

The error for this approximation has been reduced to -0.01807. ■

To generalize this procedure for an arbitrary integral $\int_a^b f(x)\,dx$, choose an even integer n. Subdivide the interval $[a, b]$ into n subintervals, and apply Simpson's rule on each consecutive pair of subintervals. (See Figure 4.7.)

Figure 4.7



Let $f \in C^4[a,b]$, n be even, $h = (b-a)/n$, and $x_j = a + jh$, for each $j = 0, 1, \ldots, n$. There exists a $\mu \in (a,b)$ for which the **Composite Simpson's rule** for n subintervals can be written with its error term as

$$\int_a^b f(x)\,dx = \frac{h}{3}\left[f(a) + 2\sum_{j=1}^{(n/2)-1} f(x_{2j}) + 4\sum_{j=1}^{n/2} f(x_{2j-1}) + f(b)\right] - \frac{b-a}{180}h^4 f^{(4)}(\mu).$$

■

2. Simpsons $\frac{1}{3}$ Rule

$$\int y\,dx = \frac{h}{3}\left(\underset{a}{y_0} + 4\left(\underbrace{y_1 + y_3 + y_5 + \ldots + \; + y_{n-1}}_{\text{odd}}\right) + 2\left(y_2 + y_4 + \underset{\text{Even}}{y_6} + \ldots + y_{n-2}\right) + \underset{b}{y_n}\right)$$

Example: Evaluate $\int_0^1 e^x dx$, by Simpson's ⅓ rule.

Solution:

Let us divide the range [0, 1] into six equal parts by taking h = 1/6.

If $x_0 = 0$ then $y_0 = e^0 = 1$.

If $x_1 = x_0 + h = ⅙$, then $y_1 = e^{1/6} = 1.1813$

If $x_2 = x_0 + 2h = 2/6 = 1/3$ then, $y_2 = e^{1/3} = 1.3956$

If $x_3 = x_0 + 3h = 3/6 = ½$ then $y_3 = e^{1/2} = 1.6487$

If $x_4 = x_0 + 4h = 4/6$ ⅔ then $y_4 = e^{2/3} = 1.9477$

If $x_5 = x_0 + 5h = ⅚$ then $y_5 = e^{5/6} = 2.3009$

If $x_6 = x_0 + 6h = 6/6 = 1$ then $y_6 = e^1 = 2.7182$

We know by Simpson's ⅓ rule;

$\int_a^b f(x)\, dx = h/3\, [(y_0 + y_n) + 4(y_1 + y_3 + y_5 + \ldots + y_{n-1}) + 2(y_2 + y_4 + y_6 + \ldots + y_{n-2})]$

Therefore,

$\int_0^1 e^x dx = (1/18)\, [(1 + 2.7182) + 4(1.1813 + 1.6487 + 2.3009) + 2(1.39561 + 1.9477)]$

$= (1/18)[3.7182 + 20.5236 + 6.68662]$

$= 1.7182$ (approx.)

Theorem 4.5 Let $f \in C^2[a,b]$, $h = (b-a)/n$, and $x_j = a + jh$, for each $j = 0, 1, \ldots, n$. There exists a $\mu \in (a,b)$ for which the **Composite Trapezoidal rule** for n subintervals can be written with its error term as

$$\int_a^b f(x)\,dx = \frac{h}{2}\left[f(a) + 2\sum_{j=1}^{n-1} f(x_j) + f(b)\right] - \frac{b-a}{12}h^2 f''(\mu).$$ ■

$[a, b]$

1. Trapezoidal Rule

$$\int y\,dx = \frac{h}{2}\left(y_0 + 2\left(y_1 + y_2 + y_3 + \ldots + y_{n-1}\right) + y_n\right)$$

a b

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Approximate the integral $\int_0^1 e^x \, dx$ with $n = 4$ using the trapezoidal rule.

$$f(x_0) = f(0) = 1$$

$$2f(x_1) = 2f\left(\frac{1}{4}\right) = 2e^{\frac{1}{4}} \approx 2.568050833375483$$

$$2f(x_2) = 2f\left(\frac{1}{2}\right) = 2e^{\frac{1}{2}} \approx 3.297442541400256$$

$$2f(x_3) = 2f\left(\frac{3}{4}\right) = 2e^{\frac{3}{4}} \approx 4.234000033225349$$

$$f(x_4) = f(1) = e \approx 2.718281828459045$$

Finally, just sum up the above values and multiply by $\frac{\Delta x}{2} = \frac{1}{8}$:

$\frac{1}{8}(1 + 2.568050833375483 + 3.297442541400256 + 4.234000033225349 + 2.7182818$
$1.727221904557517.$

Do Analysis on Exact Error of both Methods and write in one line sentence

$$\int_0^1 e^x \, dx$$

1.718281828459045

ALGORITHM
4.1

Composite Simpson's Rule

To approximate the integral $I = \int_a^b f(x)\,dx$:

INPUT endpoints a, b; even positive integer n.

OUTPUT approximation XI to I.

Step 1 Set $h = (b - a)/n$.

Step 2 Set $XI0 = f(a) + f(b)$;
$XI1 = 0$; *(Summation of $f(x_{2i-1})$.)*
$XI2 = 0$. *(Summation of $f(x_{2i})$.)*

Step 3 For $i = 1, \ldots, n - 1$ do Steps 4 and 5.

Step 4 Set $X = a + ih$.

Step 5 If i is even then set $XI2 = XI2 + f(X)$
else set $XI1 = XI1 + f(X)$.

Step 6 Set $XI = h(XI0 + 2 \cdot XI2 + 4 \cdot XI1)/3$.

Step 7 OUTPUT (XI);
STOP.

Section 4.4

Composite Trapezoidal and Composite Simpson rule covered in

1. Use the Composite Trapezoidal rule with the indicated values of n to approximate the following integrals.

a. $\int_1^2 x \ln x \, dx, \quad n = 4$

b. $\int_{-2}^{2} x^3 e^x \, dx, \quad n = 4$

c. $\int_0^2 \frac{2}{x^2+4} \, dx, \quad n = 6$

d. $\int_0^{\pi} x^2 \cos x \, dx, \quad n = 6$

e. $\int_0^2 e^{2x} \sin 3x \, dx, \quad n = 8$

f. $\int_1^3 \frac{x}{x^2+4} \, dx, \quad n = 8$

g. $\int_3^5 \frac{1}{\sqrt{x^2-4}} \, dx, \quad n = 8$

h. $\int_0^{3\pi/8} \tan x \, dx, \quad n = 8$

2. Use the Composite Trapezoidal rule with the indicated values of n to approximate the following integrals.

a. $\int_{-0.5}^{0.5} \cos^2 x \, dx, \quad n = 4$

b. $\int_{-0.5}^{0.5} x \ln(x+1) \, dx, \quad n = 6$

c. $\int_{.75}^{1.75} (\sin^2 x - 2x \sin x + 1) \, dx, \quad n = 8$

d. $\int_e^{e+2} \frac{1}{x \ln x} \, dx, \quad n = 8$

3. Use the Composite Simpson's rule to approximate the integrals in Exercise 1.

4. Use the Composite Simpson's rule to approximate the integrals in Exercise 2.

National University
of computer and emerging sciences

Foundation for Advancement of Science and Technology

Lecture # 27

Euler's Method

Euler's Method

Euler's method is the most elementary approximation technique for solving initial-value problems. Although it is seldom used in practice, the simplicity of its derivation can be used to illustrate the techniques involved in the construction of some of the more advanced techniques, without the cumbersome algebra that accompanies these constructions.

The object of Euler's method is to obtain approximations to the well-posed initial-value problem

$$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha. \tag{5.6}$$

$$y(t_{i+1}) = y(t_i) + hf(t_i, y(t_i)) + \frac{h^2}{2} y''(\xi_i).$$

The use of elementary difference methods to approximate the solution to differential equations was one of the numerous mathematical topics that was first presented to the mathematical public by the most prolific of mathematicians, Leonhard Euler (1707–1783).

In Example 1 we will use an algorithm for Euler's method to approximate the solution to

$$y' = y - t^2 + 1, \quad \boxed{0 \le t \le 2,} \quad y(0) = 0.5, \quad \text{IV}$$

at $t = 2$. Here we will simply illustrate the steps in the technique when we have $h = 0.5$.

In Example 1 we will use an algorithm for Euler's method to approximate the solution to

$$y' = y - t^2 + 1, \quad \boxed{0 \le t \le 2,} \quad y(0) = 0.5, \longrightarrow \text{IV}$$

at $t = 2$. Here we will simply illustrate the steps in the technique when we have $h = 0.5$.

For this problem $f(t, y) = y - t^2 + 1$, so

$$w_0 = y(0) = 0.5;$$

$$y(0.5) \approx w_1 = w_0 + 0.5\left(w_0 - (0.0)^2 + 1\right) = 0.5 + 0.5(1.5) = 1.25;$$

$$y(1) \approx w_2 = w_1 + 0.5\left(w_1 - (0.5)^2 + 1\right) = 1.25 + 0.5(2.0) = 2.25;$$

$$y(1.5) \approx w_3 = w_2 + 0.5\left(w_2 - (1.0)^2 + 1\right) = 2.25 + 0.5(2.25) = 3.375;$$

and

$$y(2) \approx w_4 = w_3 + 0.5\left(w_3 - (1.5)^2 + 1\right) = 3.375 + 0.5(2.125) = 4.4375.$$

ALGORITHM 5.1

Euler's

To approximate the solution of the initial-value problem

$$y' = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha,$$

at $(N + 1)$ equally spaced numbers in the interval $[a, b]$:

INPUT endpoints a, b; integer N; initial condition α.

OUTPUT approximation w to y at the $(N + 1)$ values of t.

Step 1 Set $h = (b - a)/N$;
  $t = a$;
  $w = \alpha$;
  OUTPUT (t, w).

Step 2 For $i = 1, 2, \ldots, N$ do Steps 3, 4.

  Step 3 Set $w = w + hf(t, w)$; (*Compute* w_i.)
    $t = a + ih$. (*Compute* t_i.)

  Step 4 OUTPUT (t, w).

Step 5 STOP.

EXERCISE SET 5.2

1. Use Euler's method to approximate the solutions for each of the following initial-value problems.
   a. $y' = te^{3t} - 2y, \quad 0 \le t \le 1, \quad y(0) = 0$, with $h = 0.5$
   b. $y' = 1 + (t - y)^2, \quad 2 \le t \le 3, \quad y(2) = 1$, with $h = 0.5$
   c. $y' = 1 + y/t, \quad 1 \le t \le 2, \quad y(1) = 2$, with $h = 0.25$
   d. $y' = \cos 2t + \sin 3t, \quad 0 \le t \le 1, \quad y(0) = 1$, with $h = 0.25$

2. Use Euler's method to approximate the solutions for each of the following initial-value problems.
   a. $y' = e^{t-y}, \quad 0 \le t \le 1, \quad y(0) = 1$, with $h = 0.5$
   b. $y' = \dfrac{1+t}{1+y}, \quad 1 \le t \le 2, \quad y(1) = 2$, with $h = 0.5$
   c. $y' = -y + ty^{1/2}, \quad 2 \le t \le 3, \quad y(2) = 2$, with $h = 0.25$
   d. $y' = t^{-2}(\sin 2t - 2ty), \quad 1 \le t \le 2, \quad y(1) = 2$, with $h = 0.25$

Lecture # 29

RUNGE-KUTTA Method

Taylor's Series

$$
\begin{aligned}
f(x) &= f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 \\
&\quad + \frac{f'''(x_0)}{3!}(x - x_0)^3 + \frac{f''''(x_0)}{4!}(x - x_0)^4 + \cdots \\
&= \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n.
\end{aligned}
$$

RUNGE-KUTTA METHODS

Runge-Kutta (RK) methods achieve the accuracy of a Taylor series approach without requiring the calculation of higher derivatives. Many variations exist but all can be cast in the generalized form of Eq. (22.4):

$$y_{i+1} = y_i + \phi h \tag{22.33}$$

where ϕ is called an *increment function,* which can be interpreted as a representative slope over the interval. The increment function can be written in general form as

$$\phi = a_1 k_1 + a_2 k_2 + \cdots + a_n k_n \tag{22.34}$$

where the a's are constants and the k's are

$$k_1 = f(t_i, y_i) \tag{22.34a}$$

$$k_2 = f(t_i + p_1 h, y_i + q_{11} k_1 h) \tag{22.34b}$$

$$k_3 = f(t_i + p_2 h, y_i + q_{21} k_1 h + q_{22} k_2 h) \tag{22.34c}$$

$$\vdots$$

$$k_n = f(t_i + p_{n-1} h, y_i + q_{n-1,1} k_1 h + q_{n-1,2} k_2 h + \cdots + q_{n-1,n-1} k_{n-1} h) \tag{22.34d}$$

22.4.1 Second-Order Runge-Kutta Methods

The second-order version of Eq. (22.33) is

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2)h \tag{22.35}$$

where

$$k_1 = f(t_i, y_i) \tag{22.35a}$$

$$k_2 = f(t_i + p_1 h, y_i + q_{11} k_1 h) \tag{22.35b}$$

The values for a_1, a_2, p_1, and q_{11} are evaluated by setting Eq. (22.35) equal to a second-order Taylor series. By doing this, three equations can be derived to evaluate the four unknown constants (see Chapra and Canale, 2010, for details). The three equations are

$$a_1 + a_2 = 1 \tag{22.36}$$

$$a_2 p_1 = 1/2 \tag{22.37}$$

$$a_2 q_{11} = 1/2 \tag{22.38}$$

$$a_1 = 1 - a_2$$

$$p_1 = q_{11} = \frac{1}{2a_2}$$

Heun Method without Iteration ($a_2 = 1/2$). If a_2 is assumed to be 1/2, Eqs. (22.39) and (22.40) can be solved for $a_1 = 1/2$ and $p_1 = q_{11} = 1$. These parameters, when substituted into Eq. (22.35), yield

$$y_{i+1} = y_i + \left(\frac{1}{2}k_1 + \frac{1}{2}k_2\right)h \tag{22.41}$$

where

$$k_1 = f(t_i, y_i) \tag{22.41a}$$

$$k_2 = f(t_i + h, y_i + k_1 h) \tag{22.41b}$$

Note that k_1 is the slope at the beginning of the interval and k_2 is the slope at the end of the interval. Consequently, this second-order RK method is actually Heun's technique without iteration of the corrector.

The Midpoint Method ($a_2 = 1$). If a_2 is assumed to be 1, then $a_1 = 0$, $p_1 = q_{11} = 1/2$, and Eq. (22.35) becomes

$$y_{i+1} = y_i + k_2 h \tag{22.42}$$

where

$$k_1 = f(t_i, y_i) \tag{22.42a}$$

$$k_2 = f(t_i + h/2, y_i + k_1 h/2) \tag{22.42b}$$

This is the midpoint method.

22.4.2 Classical Fourth-Order Runge-Kutta Method

The most popular RK methods are fourth order. As with the second-order approaches, there are an infinite number of versions. The following is the most commonly used form, and we therefore call it the *classical fourth-order RK method:*

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h \tag{22.44}$$

where

$$k_1 = f(t_i, y_i) \tag{22.44a}$$

$$k_2 = f\left(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1 h\right) \tag{22.44b}$$

$$k_3 = f\left(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2 h\right) \tag{22.44c}$$

$$k_4 = f(t_i + h, y_i + k_3 h) \tag{22.44d}$$

Notice that for ODEs that are a function of t alone, the classical fourth-order RK method is similar to Simpson's 1/3 rule. In addition, the fourth-order RK method is similar to the Heun approach in that multiple estimates of the slope are developed to come up with an improved average slope for the interval. As depicted in Fig. 22.7, each of the k's represents a slope. Equation (22.44) then represents a weighted average of these to arrive at the improved slope.

Problem Statement. Employ the classical fourth-order RK method to integrate $y' = 4e^{0.8t} - 0.5y$ from $t = 0$ to 1 using a step size of 1 with $y(0) = 2$.

Solution. For this case, the slope at the beginning of the interval is computed as

$$k_1 = f(0, 2) = 4e^{0.8(0)} - 0.5(2) = 3$$

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h$$

This value is used to compute a value of y and a slope at the midpoint:

$$y(0.5) = 2 + 3(0.5) = 3.5$$
$$k_2 = f(0.5, 3.5) = 4e^{0.8(0.5)} - 0.5(3.5) = 4.217299$$

This slope in turn is used to compute another value of y and another slope at the midpoint:

$$y(0.5) = 2 + 4.217299(0.5) = 4.108649$$
$$k_3 = f(0.5, 4.108649) = 4e^{0.8(0.5)} - 0.5(4.108649) = 3.912974$$

Next, this slope is used to compute a value of y and a slope at the end of the interval:

$$y(1.0) = 2 + 3.912974(1.0) = 5.912974$$
$$k_4 = f(1.0, 5.912974) = 4e^{0.8(1.0)} - 0.5(5.912974) = 5.945677$$

Finally, the four slope estimates are combined to yield an average slope. This average slope is then used to make the final prediction at the end of the interval.

$$\phi = \frac{1}{6}[3 + 2(4.217299) + 2(3.912974) + 5.945677] = 4.201037$$
$$y(1.0) = 2 + 4.201037(1.0) = 6.201037$$

which compares favorably with the true solution of 6.194631 ($\varepsilon_t = 0.103\%$).

Lecture # 33

Matrix Factorization
(LU Decomposition)

- Two basic classes of solving methods: direct and iterative

- *Direct methods* : assume that the exact solutions exists and find the precise solution in a final number of steps
  - Gauss-Jordan method (simple and accurate)
  - Gaussian elimination method (calculatively efficient)
  - Cholesky method (in case of nonsingular symmetrical matrix)

- *Iterative methods :* starting from some initial approximation value, construct a series of solution approximations such that it converges to the exact solution of a system
  - Jacobi method
  - Gauss-Seidel method
  - SOR (Successive Over-Relaxation) method

Suppose that A has been factored into the triangular form $A = LU$, where L is lower triangular and U is upper triangular. Then we can solve for $\mathbf{x}$ more easily by using a two-step process.

- First we let $\mathbf{y} = U\mathbf{x}$ and solve the lower triangular system $L\mathbf{y} = \mathbf{b}$ for $\mathbf{y}$. Since L is triangular, determining $\mathbf{y}$ from this equation requires only $O(n^2)$ operations.
- Once $\mathbf{y}$ is known, the upper triangular system $U\mathbf{x} = \mathbf{y}$ requires only an additional $O(n^2)$ operations to determine the solution $\mathbf{x}$.

Solving a linear system $A\mathbf{x} = \mathbf{b}$ in factored form means that the number of operations needed to solve the system $A\mathbf{x} = \mathbf{b}$ is reduced from $O(n^3/3)$ to $O(2n^2)$.

Example 1 Compare the approximate number of operations required to determine the solution to a linear system using a technique requiring $O(n^3/3)$ operations and one requiring $O(2n^2)$ when $n = 20$, $n = 100$, and $n = 1000$.

Solution Table 6.3 gives the results of these calculations. ■

Table 6.3

n	$n^3/3$	$2n^2$	% Reduction
10	$3.\overline{3} \times 10^2$	2×10^2	40
100	$3.\overline{3} \times 10^5$	2×10^4	94
1000	$3.\overline{3} \times 10^8$	2×10^6	99.4

As the example illustrates, the reduction factor increases dramatically with the size of the matrix. Not surprisingly, the reductions from the factorization come at a cost; determining the specific matrices L and U requires $O(n^3/3)$ operations. But once the factorization is determined, systems involving the matrix A can be solved in this simplified manner for any number of vectors $\mathbf{b}$.

Example 2 **(a)** Determine the LU factorization for matrix A in the linear system $A\mathbf{x} = \mathbf{b}$, where

$$A = \begin{bmatrix} 1 & 1 & 0 & 3 \\ 2 & 1 & -1 & 1 \\ 3 & -1 & -1 & 2 \\ -1 & 2 & 3 & -1 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ -3 \\ 4 \end{bmatrix}.$$

Solution **(a)** The original system was considered in Section 6.1, where we saw that the sequence of operations $(E_2 - 2E_1) \to (E_2)$, $(E_3 - 3E_1) \to (E_3)$, $(E_4 - (-1)E_1) \to (E_4)$, $(E_3 - 4E_2) \to (E_3)$, $(E_4 - (-3)E_2) \to (E_4)$ converts the system to the triangular system

$$\begin{aligned} x_1 + x_2 \qquad\quad + \;\; 3x_4 &= \quad 4, \\ -x_2 - \;\; x_3 - \;\; 5x_4 &= -7, \\ 3x_3 + 13x_4 &= \quad 13, \\ -13x_4 &= -13. \end{aligned}$$

The multipliers m_{ij} and the upper triangular matrix produce the factorization

$$A = \begin{bmatrix} 1 & 1 & 0 & 3 \\ 2 & 1 & -1 & 1 \\ 3 & -1 & -1 & 2 \\ -1 & 2 & 3 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 4 & 1 & 0 \\ -1 & -3 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 3 \\ 0 & -1 & -1 & -5 \\ 0 & 0 & 3 & 13 \\ 0 & 0 & 0 & -13 \end{bmatrix} = LU.$$

(b) Then use the factorization to solve the system

$$\begin{aligned} x_1 + x_2 \phantom{{}- x_3} + 3x_4 &= 8, \\ 2x_1 + x_2 - x_3 + x_4 &= 7, \\ 3x_1 - x_2 - x_3 + 2x_4 &= 14, \\ -x_1 + 2x_2 + 3x_3 - x_4 &= -7. \end{aligned}$$

$$A\mathbf{x} = LU\mathbf{x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 4 & 1 & 0 \\ -1 & -3 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 3 \\ 0 & -1 & -1 & -5 \\ 0 & 0 & 3 & 13 \\ 0 & 0 & 0 & -13 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 8 \\ 7 \\ 14 \\ -7 \end{bmatrix},$$

we first introduce the substitution $\mathbf{y} = U\mathbf{x}$. Then $\mathbf{b} = L(U\mathbf{x}) = L\mathbf{y}$. That is,

$$L\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 4 & 1 & 0 \\ -1 & -3 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 8 \\ 7 \\ 14 \\ -7 \end{bmatrix}.$$

This system is solved for $\mathbf{y}$ by a simple forward-substitution process:

$$\begin{aligned} y_1 &= 8; \\ 2y_1 + y_2 &= 7, \quad \text{so } y_2 = 7 - 2y_1 = -9; \\ 3y_1 + 4y_2 + y_3 &= 14, \quad \text{so } y_3 = 14 - 3y_1 - 4y_2 = 26; \\ -y_1 - 3y_2 + y_4 &= -7, \quad \text{so } y_4 = -7 + y_1 + 3y_2 = -26. \end{aligned}$$

We then solve $U\mathbf{x} = \mathbf{y}$ for $\mathbf{x}$, the solution of the original system; that is,

$$\begin{bmatrix} 1 & 1 & 0 & 3 \\ 0 & -1 & -1 & -5 \\ 0 & 0 & 3 & 13 \\ 0 & 0 & 0 & -13 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 8 \\ -9 \\ 26 \\ -26 \end{bmatrix}.$$

Using backward substitution we obtain $x_4 = 2$, $x_3 = 0$, $x_2 = -1$, $x_1 = 3$. ■

ALGORITHM 6.4

LU Factorization

To factor the $n \times n$ matrix $A = [a_{ij}]$ into the product of the lower-triangular matrix $L = [l_{ij}]$ and the upper-triangular matrix $U = [u_{ij}]$; that is, $A = LU$, where the main diagonal of either L or U consists of all ones:

INPUT dimension n; the entries a_{ij}, $1 \le i, j \le n$ of A; the diagonal $l_{11} = \cdots = l_{nn} = 1$ of L or the diagonal $u_{11} = \cdots = u_{nn} = 1$ of U.

OUTPUT the entries l_{ij}, $1 \le j \le i$, $1 \le i \le n$ of L and the entries, u_{ij}, $i \le j \le n$, $1 \le i \le n$ of U.

Step 1 Select l_{11} and u_{11} satisfying $l_{11}u_{11} = a_{11}$.
If $l_{11}u_{11} = 0$ then OUTPUT ('Factorization impossible');
STOP.

Step 2 For $j = 2, \ldots, n$ set $u_{1j} = a_{1j}/l_{11}$; *(First row of U.)*
$l_{j1} = a_{j1}/u_{11}$. *(First column of L.)*

Step 3 For $i = 2, \ldots, n-1$ do Steps 4 and 5.

Step 4 Select l_{ii} and u_{ii} satisfying $l_{ii}u_{ii} = a_{ii} - \sum_{k=1}^{i-1} l_{ik}u_{ki}$.
If $l_{ii}u_{ii} = 0$ then OUTPUT ('Factorization impossible');
STOP.

Step 5 For $j = i+1, \ldots, n$

set $u_{ij} = \frac{1}{l_{ii}}\left[a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}\right]$; *(ith row of U.)*

$l_{ji} = \frac{1}{u_{ii}}\left[a_{ji} - \sum_{k=1}^{i-1} l_{jk}u_{ki}\right]$. *(ith column of L.)*

Step 6 Select l_{nn} and u_{nn} satisfying $l_{nn}u_{nn} = a_{nn} - \sum_{k=1}^{n-1} l_{nk}u_{kn}$.
(Note: If $l_{nn}u_{nn} = 0$, then $A = LU$ but A is singular.)

Step 7 OUTPUT (l_{ij} for $j = 1, \ldots, i$ and $i = 1, \ldots, n$);
OUTPUT (u_{ij} for $j = i, \ldots, n$ and $i = 1, \ldots, n$);
STOP.

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Theorem 6.19 If Gaussian elimination can be performed on the linear system $A\mathbf{x} = \mathbf{b}$ without row interchanges, then the matrix A can be factored into the product of a lower-triangular matrix L and an upper-triangular matrix U, that is, $A = LU$, where $m_{ji} = a_{ji}^{(i)}/a_{ii}^{(i)}$,

$$U = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1,n}^{(n-1)} \\ 0 & \cdots & 0 & a_{nn}^{(n)} \end{bmatrix}, \quad \text{and} \quad L = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ m_{21} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ m_{n1} & \cdots & m_{n,n-1} & 1 \end{bmatrix}.$$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

An $n \times n$ **permutation matrix** $P = [p_{ij}]$ is a matrix obtained by rearranging the rows of I_n, the identity matrix. This gives a matrix with precisely one nonzero entry in each row and in each column, and each nonzero entry is a 1.

Recall Elementary Matrices!

Illustration The matrix

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

is a 3×3 permutation matrix. For any 3×3 matrix A, multiplying on the left by P has the effect of interchanging the second and third rows of A:

$$PA = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{31} & a_{32} & a_{33} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}.$$

- P^{-1} exists and $P^{-1} = P^t$.

$$PA\mathbf{x} = P\mathbf{b}$$

can be solved without row interchanges. As a consequence, this matrix PA can be factored into

$$PA = LU,$$

where L is lower triangular and U is upper triangular. Because $P^{-1} = P^t$, this produces the factorization

$$A = P^{-1}LU = (P^tL)U.$$

The matrix U is still upper triangular, but P^tL is not lower triangular unless $P = I$.

Example 3 Determine a factorization in the form $A = (P^tL)U$ for the matrix

$$A = \begin{bmatrix} 0 & 0 & -1 & 1 \\ 1 & 1 & -1 & 2 \\ -1 & -1 & 2 & 0 \\ 1 & 2 & 0 & 2 \end{bmatrix}.$$

Solution The matrix A cannot have an LU factorization because $a_{11} = 0$. However, using the row interchange $(E_1) \leftrightarrow (E_2)$, followed by $(E_3 + E_1) \rightarrow (E_3)$ and $(E_4 - E_1) \rightarrow (E_4)$, produces

$$\begin{bmatrix} 1 & 1 & -1 & 2 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & 2 \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

Then the row interchange $(E_2) \leftrightarrow (E_4)$, followed by $(E_4 + E_3) \rightarrow (E_4)$, gives the matrix

$$U = \begin{bmatrix} 1 & 1 & -1 & 2 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 3 \end{bmatrix}.$$

The permutation matrix associated with the row interchanges $(E_1) \leftrightarrow (E_2)$ and $(E_2) \leftrightarrow (E_4)$ is

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$

and

$$PA = \begin{bmatrix} 1 & 1 & -1 & 2 \\ 1 & 2 & 0 & 2 \\ -1 & -1 & 2 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}.$$

Gaussian elimination is performed on PA using the same operations as on A, except without the row interchanges. That is, $(E_2 - E_1) \rightarrow (E_2)$, $(E_3 + E_1) \rightarrow (E_3)$, followed by $(E_4 + E_3) \rightarrow (E_4)$. The nonzero multipliers for PA are consequently,

$$m_{21} = 1, \quad m_{31} = -1, \quad \text{and} \quad m_{43} = -1,$$

and the LU factorization of PA is

$$PA = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 & 2 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 3 \end{bmatrix} = LU.$$

Multiplying by $P^{-1} = P^t$ produces the factorization

$$A = P^{-1}(LU) = P^t(LU) = (P^tL)U = \begin{bmatrix} 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 & 2 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 3 \end{bmatrix}. \quad ■$$

Do Q 1,2,3 & 6 from Ex#6.5

Doolittle form

Obtained by

Gaussian elimination

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

Method: [A] Decomposes to [L] and [U]

$$[A]=[L][U]=\begin{bmatrix} 1 & 0 & 0 \\ \ell_{21} & 1 & 0 \\ \ell_{31} & \ell_{32} & 1 \end{bmatrix}\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

[*U*] is the same as the coefficient matrix at the end of the forward elimination step.

[*L*] is obtained using the *multipliers* that were used in the forward elimination process

Finding the [*U*] matrix

Using the Forward Elimination Procedure of Gauss Elimination

$$\begin{bmatrix} 25 & 5 & 1 \\ 64 & 8 & 1 \\ 144 & 12 & 1 \end{bmatrix}$$

Step 1: $$\frac{64}{25} = 2.56; \quad Row2 - Row1(2.56) = \begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 144 & 12 & 1 \end{bmatrix}$$

$$\frac{144}{25} = 5.76; \quad Row3 - Row1(5.76) = \begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 0 & -16.8 & -4.76 \end{bmatrix}$$

Finding the [U] Matrix

Matrix after Step 1:
$$\begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 0 & -16.8 & -4.76 \end{bmatrix}$$

Step 2:
$$\frac{-16.8}{-4.8} = 3.5; \quad Row3 - Row2(3.5) = \begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 0 & 0 & 0.7 \end{bmatrix}$$

$$[U] = \begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 0 & 0 & 0.7 \end{bmatrix}$$

Finding the [L] matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ \ell_{21} & 1 & 0 \\ \ell_{31} & \ell_{32} & 1 \end{bmatrix}$$

Using the multipliers used during the Forward Elimination Procedure

From the first step of forward elimination

$$\begin{bmatrix} 25 & 5 & 1 \\ 64 & 8 & 1 \\ 144 & 12 & 1 \end{bmatrix}$$

$$\ell_{21} = \frac{a_{21}}{a_{11}} = \frac{64}{25} = 2.56$$

$$\ell_{31} = \frac{a_{31}}{a_{11}} = \frac{144}{25} = 5.76$$

Finding the [L] Matrix

From the second step of forward elimination

$$\begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 0 & -16.8 & -4.76 \end{bmatrix} \quad \ell_{32} = \frac{a_{32}}{a_{22}} = \frac{-16.8}{-4.8} = 3.5$$

$$[L] = \begin{bmatrix} 1 & 0 & 0 \\ 2.56 & 1 & 0 \\ 5.76 & 3.5 & 1 \end{bmatrix}$$

Does [L][U] = [A]?

$$[L][U]=\begin{bmatrix}1 & 0 & 0\\ 2.56 & 1 & 0\\ 5.76 & 3.5 & 1\end{bmatrix}\begin{bmatrix}25 & 5 & 1\\ 0 & -4.8 & -1.56\\ 0 & 0 & 0.7\end{bmatrix}= \quad ?$$

Using LU Decomposition to solve SLEs

Solve the following set of linear equations using LU Decomposition

$$\begin{bmatrix} 25 & 5 & 1 \\ 64 & 8 & 1 \\ 144 & 12 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 106.8 \\ 177.2 \\ 279.2 \end{bmatrix}$$

Using the procedure for finding the [*L*] and [*U*] matrices

$$[A] = [L][U] = \begin{bmatrix} 1 & 0 & 0 \\ 2.56 & 1 & 0 \\ 5.76 & 3.5 & 1 \end{bmatrix} \begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 0 & 0 & 0.7 \end{bmatrix}$$

Example

Set $[L][y] = [b]$

$$\begin{bmatrix} 1 & 0 & 0 \\ 2.56 & 1 & 0 \\ 5.76 & 3.5 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 106.8 \\ 177.2 \\ 279.2 \end{bmatrix}$$

Solve for $[y]$

$$y_1 = 10$$
$$2.56y_1 + y_2 = 177.2$$
$$5.76y_1 + 3.5y_2 + y_3 = 279.2$$

Example

Complete the forward substitution to solve for $[Z]$

$$
\begin{aligned}
y_1 &= 106.8 \\
y_2 &= 177.2 - 2.56 y_1 \\
&= 177.2 - 2.56(106.8) \\
&= -96.2 \\
y_3 &= 279.2 - 5.76 y_1 - 3.5 y_2 \\
&= 279.2 - 5.76(106.8) - 3.5(-96.21) \\
&= 0.735
\end{aligned}
$$

$$
[y] = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 106.8 \\ -96.21 \\ 0.735 \end{bmatrix}
$$

Example

Set $[U][X]=[y]$

$$\begin{bmatrix} 25 & 5 & 1 \\ 0 & -4.8 & -1.56 \\ 0 & 0 & 0.7 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} 106.8 \\ -96.21 \\ 0.735 \end{bmatrix}$$

Solve for $[X]$

The 3 equations become

$$\begin{aligned} 25x_1+5x_2+x_3 &= 106.8 \\ -4.8x_2-1.56x_3 &= -96.21 \\ 0.7x_3 &= 0.735 \end{aligned}$$

Example

From the 3rd equation

$$0.7x_3 = 0.735$$

$$x_3 = \frac{0.735}{0.7}$$

$$x_3 = 1.050$$

Substituting in a_3 and using the second equation

$$-4.8x_2 - 1.56x_3 = -96.21$$

$$x_2 = \frac{-96.21 + 1.56x_3}{-4.8}$$

$$x_2 = \frac{-96.21 + 1.56(1.050)}{-4.8}$$

$$x_2 = 19.70$$

Example

Substituting in a_3 and a_2 using the first equation

$$25x_1 + 5x_2 + x_3 = 106.8$$

$$\begin{aligned} x_1 &= \frac{106.8 - 5a_2 - a_3}{25} \\ &= \frac{106.8 - 5(19.70) - 1.050}{25} \\ &= 0.2900 \end{aligned}$$

Hence the Solution Vector is:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.2900 \\ 19.70 \\ 1.050 \end{bmatrix}$$

Lecture # 38-40

Matrix Factorization (LU Decomposition)

Special Types of Matrices

1. Diagonally Dominant Matrices

Each main diagonal entry in a strictly diagonally dominant matrix has a magnitude that is strictly greater that the sum of the magnitudes of all the other entries in that row.

1. Diagonally Dominant Matrices

The $n \times n$ matrix A is said to be **diagonally dominant** when

$$|a_{ii}| \geq \sum_{\substack{j=1, \\ j \neq i}}^{n} |a_{ij}| \quad \text{holds for each } i = 1, 2, \cdots, n. \tag{6.10}$$

A diagonally dominant matrix is said to be **strictly diagonally dominant** when the inequality in (6.10) is strict for each n, that is, when

$$|a_{ii}| > \sum_{\substack{j=1, \\ j \neq i}}^{n} |a_{ij}| \quad \text{holds for each } i = 1, 2, \cdots, n. \quad \blacksquare$$

Consider the matrices

$$A = \begin{bmatrix} 7 & 2 & 0 \\ 3 & 5 & -1 \\ 0 & 5 & -6 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 6 & 4 & -3 \\ 4 & -2 & 0 \\ -3 & 0 & 1 \end{bmatrix}.$$

The nonsymmetric matrix A is strictly diagonally dominant because

$$|7| > |2| + |0|, \quad |5| > |3| + |-1|, \quad \text{and} \quad |-6| > |0| + |5|.$$

The symmetric matrix B is not strictly diagonally dominant because, for example, in the first row the absolute value of the diagonal element is $|6| < |4| + |-3| = 7$. It is interesting to note that A^t is not strictly diagonally dominant, because the middle row of A^t is $[2\ 5\ 5]$, nor, of course, is B^t because $B^t = B$. □

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Theorem 6.21 A strictly diagonally dominant matrix A is nonsingular. Moreover, in this case, Gaussian elimination can be performed on any linear system of the form $A\mathbf{x} = \mathbf{b}$ to obtain its unique solution without row or column interchanges, and the computations will be stable with respect to the growth of round-off errors. ■

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

2. Positive Definite Matrices

The name positive definite refers to the fact that the number $\mathbf{x}'A\mathbf{x}$ must be positive whenever $\mathbf{x} \neq \mathbf{0}$.

2. Positive Definite Matrices

Definition 6.22 A matrix A is **positive definite** if it is symmetric and if $\mathbf{x}'A\mathbf{x} > 0$ for every n-dimensional vector $\mathbf{x} \neq \mathbf{0}$. ■

Example 1 Show that the matrix

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

is positive definite

Solution Suppose $\mathbf{x}$ is any three-dimensional column vector. Then

$$\begin{aligned}
\mathbf{x}'A\mathbf{x} &= [x_1, x_2, x_3] \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \\
&= [x_1, x_2, x_3] \begin{bmatrix} 2x_1 & - & x_2 & & \\ -x_1 & + & 2x_2 & - & x_3 \\ -x_2 & + & 2x_3 & & \end{bmatrix} \\
&= 2x_1^2 - 2x_1x_2 + 2x_2^2 - 2x_2x_3 + 2x_3^2.
\end{aligned}$$

$$= 2x_1^2 - 2x_1x_2 + 2x_2^2 - 2x_2x_3 + 2x_3^2.$$

Rearranging the terms gives

$$\begin{aligned} \mathbf{x}'A\mathbf{x} &= x_1^2 + (x_1^2 - 2x_1x_2 + x_2^2) + (x_2^2 - 2x_2x_3 + x_3^2) + x_3^2 \\ &= x_1^2 + (x_1 - x_2)^2 + (x_2 - x_3)^2 + x_3^2, \end{aligned}$$

which implies that

$$x_1^2 + (x_1 - x_2)^2 + (x_2 - x_3)^2 + x_3^2 > 0$$

unless $x_1 = x_2 = x_3 = 0$.

Theorem 6.23 If A is an $n \times n$ positive definite matrix, then

(i) A has an inverse; **(ii)** $a_{ii} > 0$, for each $i = 1, 2, \ldots, n$;

(iii) $\max_{1 \le k, j \le n} |a_{kj}| \le \max_{1 \le i \le n} |a_{ii}|$; **(iv)** $(a_{ij})^2 < a_{ii}a_{jj}$, for each $i \ne j$. ■

Definition 6.24 A **leading principal submatrix** of a matrix A is a matrix of the form

$$A_k = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{bmatrix},$$

for some $1 \le k \le n$. ■

Theorem 6.25 A symmetric matrix A is positive definite if and only if each of its leading principal submatrices has a positive determinant. ■

Example 2 In Example 1 we used the definition to show that the symmetric matrix

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

is positive definite. Confirm this using Theorem 6.25.

Solution Note that

$$\det A_1 = \det[2] = 2 > 0,$$

$$\det A_2 = \det \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} = 4 - 1 = 3 > 0,$$

and

$$\det A_3 = \det \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} = 2 \det \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} - (-1) \det \begin{bmatrix} -1 & -1 \\ 0 & 2 \end{bmatrix}$$

$$= 2(4 - 1) + (-2 + 0) = 4 > 0.$$

in agreement with Theorem 6.25. ■

Theorem 6.26 The symmetric matrix A is positive definite if and only if Gaussian elimination without row interchanges can be performed on the linear system $A\mathbf{x} = \mathbf{b}$ with all pivot elements positive. Moreover, in this case, the computations are stable with respect to the growth of round-off errors. ■

Corollary 6.27 The matrix A is positive definite if and only if A can be factored in the form LDL^t, where L is lower triangular with 1s on its diagonal and D is a diagonal matrix with positive diagonal entries. ■

Corollary 6.28 The matrix A is positive definite if and only if A can be factored in the form LL^t, where L is lower triangular with nonzero diagonal entries. ■

LDL^t Factorization

Corollary 6.29 Let A be a symmetric $n \times n$ matrix for which Gaussian elimination can be applied without row interchanges. Then A can be factored into LDL^t, where L is lower triangular with 1s on its diagonal and D is the diagonal matrix with $a_{11}^{(1)}, \ldots, a_{nn}^{(n)}$ on its diagonal. ■

Example 3 Determine the LDL^t factorization of the positive definite matrix

$$A = \begin{bmatrix} 4 & -1 & 1 \\ -1 & 4.25 & 2.75 \\ 1 & 2.75 & 3.5 \end{bmatrix}.$$

Example 3 Determine the LDL^t factorization of the positive definite matrix

$$A = \begin{bmatrix} 4 & -1 & 1 \\ -1 & 4.25 & 2.75 \\ 1 & 2.75 & 3.5 \end{bmatrix}.$$

Solution The LDL^t factorization has 1s on the diagonal of the lower triangular matrix L so we need to have

$$A = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{21} & a_{22} & a_{32} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix} \begin{bmatrix} 1 & l_{21} & l_{31} \\ 0 & 1 & l_{32} \\ 0 & 0 & 1 \end{bmatrix}$$

Solution The LDL^t factorization has 1s on the diagonal of the lower triangular matrix L so we need to have

$$A = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{21} & a_{22} & a_{32} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix} \begin{bmatrix} 1 & l_{21} & l_{31} \\ 0 & 1 & l_{32} \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 4 & -1 & 1 \\ -1 & 4.25 & 2.75 \\ 1 & 2.75 & 3.5 \end{bmatrix} = \begin{bmatrix} d_1 & d_1 l_{21} & d_1 l_{31} \\ d_1 l_{21} & d_2 + d_1 l_{21}^2 & d_2 l_{32} + d_1 l_{21} l_{31} \\ d_1 l_{31} & d_1 l_{21} l_{31} + d_2 l_{32} & d_1 l_{31}^2 + d_2 l_{32}^2 + d_3 \end{bmatrix}$$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Thus

$$a_{11}: 4 = d_1 \implies d_1 = 4, \qquad a_{21}: -1 = d_1 l_{21} \implies l_{21} = -0.25$$

$$a_{31}: 1 = d_1 l_{31} \implies l_{31} = 0.25, \qquad a_{22}: 4.25 = d_2 + d_1 l_{21}^2 \implies d_2 = 4$$

$$a_{32}: 2.75 = d_1 l_{21} l_{31} + d_2 l_{32} \implies l_{32} = 0.75, \quad a_{33}: 3.5 = d_1 l_{31}^2 + d_2 l_{32}^2 + d_3 \implies d_3 = 1,$$

Thus

$$a_{11} : 4 = d_1 \Longrightarrow d_1 = 4, \qquad a_{21} : -1 = d_1 l_{21} \Longrightarrow l_{21} = -0.25$$

$$a_{31} : 1 = d_1 l_{31} \Longrightarrow l_{31} = 0.25, \qquad a_{22} : 4.25 = d_2 + d_1 l_{21}^2 \Longrightarrow d_2 = 4$$

$$a_{32} : 2.75 = d_1 l_{21} l_{31} + d_2 l_{32} \Longrightarrow l_{32} = 0.75, \quad a_{33} : 3.5 = d_1 l_{31}^2 + d_2 l_{32}^2 + d_3 \Longrightarrow d_3 = 1,$$

and we have

$$A = LDL^t = \begin{bmatrix} 1 & 0 & 0 \\ -0.25 & 1 & 0 \\ 0.25 & 0.75 & 1 \end{bmatrix} \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -0.25 & 0.25 \\ 0 & 1 & 0.75 \\ 0 & 0 & 1 \end{bmatrix}. \qquad \blacksquare$$

ALGORITHM
6.5

LDL^t Factorization

To factor the positive definite $n \times n$ matrix A into the form LDL^t, where L is a lower triangular matrix with 1s along the diagonal and D is a diagonal matrix with positive entries on the diagonal:

INPUT the dimension n; entries a_{ij}, for $1 \le i, j \le n$ of A.

OUTPUT the entries l_{ij}, for $1 \le j < i$ and $1 \le i \le n$ of L, and d_i, for $1 \le i \le n$ of D.

Step 1 For $i = 1, \ldots, n$ do Steps 2–4.

Step 2 For $j = 1, \ldots, i-1$, set $v_j = l_{ij} d_j$.

Step 3 Set $d_i = a_{ii} - \sum_{j=1}^{i-1} l_{ij} v_j$.

Step 4 For $j = i+1, \ldots, n$ set $l_{ji} = (a_{ji} - \sum_{k=1}^{i-1} l_{jk} v_k)/d_i$.

Step 5 OUTPUT (l_{ij} for $j = 1, \ldots, i-1$ and $i = 1, \ldots, n$);
OUTPUT (d_i for $i = 1, \ldots, n$);
STOP.

Cholesky

To factor the positive definite $n \times n$ matrix A into LL^t, where L is lower triangular:

Andre-Louis Cholesky (1875-1918) was a French military officer involved in geodesy and surveying in the early 1900s. He developed this factorization method to compute solutions to least squares problems.

Cholesky

To factor the positive definite $n \times n$ matrix A into LL^t, where L is lower triangular:

INPUT the dimension n; entries a_{ij}, for $1 \le i, j \le n$ of A.

OUTPUT the entries l_{ij}, for $1 \le j \le i$ and $1 \le i \le n$ of L. (*The entries of* $U = L^t$ *are* $u_{ij} = l_{ji}$, *for* $i \le j \le n$ *and* $1 \le i \le n$.)

Step 1 Set $l_{11} = \sqrt{a_{11}}$.

Step 2 For $j = 2, \ldots, n$, set $l_{j1} = a_{j1}/l_{11}$.

Step 3 For $i = 2, \ldots, n-1$ do Steps 4 and 5.

Step 4 Set $l_{ii} = \left(a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2\right)^{1/2}$.

Step 5 For $j = i+1, \ldots, n$

set $l_{ji} = \left(a_{ji} - \sum_{k=1}^{i-1} l_{jk} l_{ik}\right) / l_{ii}$.

Step 6 Set $l_{nn} = \left(a_{nn} - \sum_{k=1}^{n-1} l_{nk}^2\right)^{1/2}$.

Step 7 OUTPUT (l_{ij} for $j = 1, \ldots, i$ and $i = 1, \ldots, n$);
STOP.

Determine the Cholesky LL^t factorization of the positive definite matrix

$$A = \begin{bmatrix} 4 & -1 & 1 \\ -1 & 4.25 & 2.75 \\ 1 & 2.75 & 3.5 \end{bmatrix}.$$

Solution The LL^t factorization does not necessarily has 1s on the diagonal of the lower triangular matrix L so we need to have

Determine the Cholesky LL^t factorization of the positive definite matrix

$$A = \begin{bmatrix} 4 & -1 & 1 \\ -1 & 4.25 & 2.75 \\ 1 & 2.75 & 3.5 \end{bmatrix}.$$

$$A = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{21} & a_{22} & a_{32} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix}$$

$$A = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{21} & a_{22} & a_{32} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix}$$

$$\begin{bmatrix} 4 & -1 & 1 \\ -1 & 4.25 & 2.75 \\ 1 & 2.75 & 3.5 \end{bmatrix} = \begin{bmatrix} l_{11}^2 & l_{11}l_{21} & l_{11}l_{31} \\ l_{11}l_{21} & l_{21}^2 + l_{22}^2 & l_{21}l_{31} + l_{22}l_{32} \\ l_{11}l_{31} & l_{21}l_{31} + l_{22}l_{32} & l_{31}^2 + l_{32}^2 + l_{33}^2 \end{bmatrix}$$

$$\begin{bmatrix} 4 & -1 & 1 \\ -1 & 4.25 & 2.75 \\ 1 & 2.75 & 3.5 \end{bmatrix} = \begin{bmatrix} l_{11}^2 & l_{11}l_{21} & l_{11}l_{31} \\ l_{11}l_{21} & l_{21}^2 + l_{22}^2 & l_{21}l_{31} + l_{22}l_{32} \\ l_{11}l_{31} & l_{21}l_{31} + l_{22}l_{32} & l_{31}^2 + l_{32}^2 + l_{33}^2 \end{bmatrix}$$

Thus

$$a_{11}: \quad 4 = l_{11}^2 \implies l_{11} = 2, \qquad a_{21}: \quad -1 = l_{11}l_{21} \implies l_{21} = -0.5$$

$$a_{31}: \quad 1 = l_{11}l_{31} \implies l_{31} = 0.5, \qquad a_{22}: \quad 4.25 = l_{21}^2 + l_{22}^2 \implies l_{22} = 2$$

$$a_{32}: \quad 2.75 = l_{21}l_{31} + l_{22}l_{32} \implies l_{32} = 1.5, \quad a_{33}: \quad 3.5 = l_{31}^2 + l_{32}^2 + l_{33}^2 \implies l_{33} = 1,$$

and we have

$$A = LL^t = \begin{bmatrix} 2 & 0 & 0 \\ -0.5 & 2 & 0 \\ 0.5 & 1.5 & 1 \end{bmatrix} \begin{bmatrix} 2 & -0.5 & 0.5 \\ 0 & 2 & 1.5 \\ 0 & 0 & 1 \end{bmatrix}. \qquad \blacksquare$$

Band Matrices

The last class of matrices considered are *band matrices*. In many applications, the band matrices are also strictly diagonally dominant or positive definite.

An $n \times n$ matrix is called a **band matrix** if integers p and q, with $1 < p, q < n$, exist with the property that $a_{ij} = 0$ whenever $p \leq j - i$ or $q \leq i - j$. The **band width** of a band matrix is defined as $w = p + q - 1$. ■

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Band Matrices

An $n \times n$ matrix is called a **band matrix** if integers p and q, with $1 < p, q < n$, exist with the property that $a_{ij} = 0$ whenever $p \leq j - i$ or $q \leq i - j$. The **band width** of a band matrix is defined as $w = p + q - 1$. ■

The number p describes the number of diagonals above, and including, the main diagonal on which nonzero entries may lie. The number q describes the number of diagonals below, and including, the main diagonal on which nonzero entries may lie. For example, the matrix

$$A = \begin{bmatrix} 7 & 2 & 0 \\ 3 & 5 & -1 \\ 0 & -5 & -6 \end{bmatrix}$$

is a band matrix with $p = q = 2$ and bandwidth $2 + 2 - 1 = 3$.

Tridiagonal Matrices

Matrices of bandwidth 3 occurring when $p = q = 2$ are called **tridiagonal** because they have the form

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & \cdots & 0 \\ a_{21} & a_{22} & a_{23} & \ddots & & \vdots \\ 0 & a_{32} & a_{33} & a_{34} & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & a_{n-1,n} \\ 0 & \cdots & \cdots & 0 & a_{n,n-1} & a_{nn} \end{bmatrix}.$$

National University
of computer and emerging sciences

Foundation for Advancement of Science and Technology

$$L = \begin{bmatrix} l_{11} & 0 & \cdots & \cdots & 0 \\ l_{21} & l_{22} & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \\ 0 & \cdots & 0 & l_{n,n-1} & l_{nn} \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} 1 & u_{12} & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ & & \ddots & \ddots & u_{n-1,n} \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}.$$

Crout Factorization for Tridiagonal Linear Systems

To solve the $n \times n$ linear system

$$
\begin{array}{llr}
E_1: & a_{11}x_1 + a_{12}x_2 & = a_{1,n+1}, \\
E_2: & a_{21}x_1 + a_{22}x_2 + a_{23}x_3 & = a_{2,n+1}, \\
\vdots & \qquad\qquad\vdots & \vdots \\
E_{n-1}: & a_{n-1,n-2}x_{n-2} + a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n & = a_{n-1,n+1}, \\
E_n: & a_{n,n-1}x_{n-1} + a_{nn}x_n & = a_{n,n+1},
\end{array}
$$

which is assumed to have a unique solution:

Determine the Crout factorization of the symmetric tridiagonal matrix

$$\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix},$$

and use this factorization to solve the linear system

$$\begin{aligned} 2x_1 - x_2 \phantom{{}+2x_3-x_4} &= 1, \\ -x_1 + 2x_2 - x_3 \phantom{{}-x_4} &= 0, \\ - x_2 + 2x_3 - x_4 &= 0, \\ - x_3 + 2x_4 &= 1. \end{aligned}$$

Solution The LU factorization of A has the form

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ 0 & a_{32} & a_{33} & a_{34} \\ 0 & 0 & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ 0 & l_{32} & l_{33} & 0 \\ 0 & 0 & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & 0 & 0 \\ 0 & 1 & u_{23} & 0 \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution The LU factorization of A has the form

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ 0 & a_{32} & a_{33} & a_{34} \\ 0 & 0 & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ 0 & l_{32} & l_{33} & 0 \\ 0 & 0 & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & 0 & 0 \\ 0 & 1 & u_{23} & 0 \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} = \begin{bmatrix} l_{11} & l_{11}u_{12} & 0 & 0 \\ l_{21} & l_{22} + l_{21}u_{12} & l_{22}u_{23} & 0 \\ 0 & l_{32} & l_{33} + l_{32}u_{23} & l_{33}u_{34} \\ 0 & 0 & l_{43} & l_{44} + l_{43}u_{34} \end{bmatrix}.$$

Thus

$$
\begin{array}{llll}
a_{11}: & 2 = l_{11} \implies l_{11} = 2, & a_{12}: & -1 = l_{11}u_{12} \implies u_{12} = -\frac{1}{2}, \\
a_{21}: & -1 = l_{21} \implies l_{21} = -1, & a_{22}: & 2 = l_{22} + l_{21}u_{12} \implies l_{22} = -\frac{3}{2}, \\
a_{23}: & -1 = l_{22}u_{23} \implies u_{23} = -\frac{2}{3}, & a_{32}: & -1 = l_{32} \implies l_{32} = -1, \\
a_{33}: & 2 = l_{33} + l_{32}u_{23} \implies l_{33} = \frac{4}{3}, & a_{34}: & -1 = l_{33}u_{34} \implies u_{34} = -\frac{3}{4}, \\
a_{43}: & -1 = l_{43} \implies l_{43} = -1, & a_{44}: & 2 = l_{44} + l_{43}u_{34} \implies l_{44} = \frac{5}{4}.
\end{array}
$$

This gives the Crout factorization

$$
A = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ -1 & \frac{3}{2} & 0 & 0 \\ 0 & -1 & \frac{4}{3} & 0 \\ 0 & 0 & -1 & \frac{5}{4} \end{bmatrix} \begin{bmatrix} 1 & -\frac{1}{2} & 0 & 0 \\ 0 & 1 & -\frac{2}{3} & 0 \\ 0 & 0 & 1 & -\frac{3}{4} \\ 0 & 0 & 0 & 1 \end{bmatrix} = LU.
$$

Solving the system

$$Lz = \begin{bmatrix} 2 & 0 & 0 & 0 \\ -1 & \frac{3}{2} & 0 & 0 \\ 0 & -1 & \frac{4}{3} & 0 \\ 0 & 0 & -1 & \frac{5}{4} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \text{gives} \quad \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{3} \\ \frac{1}{4} \\ 1 \end{bmatrix},$$

and then solving

$$U\mathbf{x} = \begin{bmatrix} 1 & -\frac{1}{2} & 0 & 0 \\ 0 & 1 & -\frac{2}{3} & 0 \\ 0 & 0 & 1 & -\frac{3}{4} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{3} \\ \frac{1}{4} \\ 1 \end{bmatrix} \quad \text{gives} \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \quad \blacksquare$$

Do the following from Book

(Ex 6.6)

Do Q 1,3,5, 11 & 12

Lecture # 41-42

The Jacobi & Gauss-Siedel Iterative Techniques

Jacobi's Method

The **Jacobi iterative method** is obtained by solving the ith equation in $A\mathbf{x} = \mathbf{b}$ for x_i to obtain (provided $a_{ii} \neq 0$)

$$x_i = \sum_{\substack{j=1 \\ j\neq i}}^{n} \left(- \frac{a_{ij}x_j}{a_{ii}} \right) + \frac{b_i}{a_{ii}}, \qquad \text{for } i = 1, 2, \ldots, n.$$

For each $k \geq 1$, generate the components $x_i^{(k)}$ of $\mathbf{x}^{(k)}$ from the components of $\mathbf{x}^{(k-1)}$ by

$$x_i^{(k)} = \frac{1}{a_{ii}} \left[\sum_{\substack{j=1 \\ j\neq i}}^{n} \left(-a_{ij}x_j^{(k-1)} \right) + b_i \right], \qquad \text{for } i = 1, 2, \ldots, n. \tag{7.5}$$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Carl Gustav Jacob Jacobi (1804–1851) was initially recognized for his work in the area of number theory and elliptic functions, but his mathematical interests and abilities were very broad. He had a strong personality that was influential in establishing a research-oriented attitude that became the nucleus of a revival of mathematics at German universities in the 19th century.

Solve:

$$\begin{aligned}
10x_1 - x_2 + 2x_3 \phantom{{}+3x_4} &= 6, \\
-x_1 + 11x_2 - x_3 + 3x_4 &= 25, \\
2x_1 - x_2 + 10x_3 - x_4 &= -11, \\
3x_2 - x_3 + 8x_4 &= 15
\end{aligned}$$

has the unique solution $\mathbf{x} = (1, 2, -1, 1)^t$. Use Jacobi's iterative technique to find approximations $\mathbf{x}^{(k)}$ to $\mathbf{x}$ starting with $\mathbf{x}^{(0)} = (0, 0, 0, 0)^t$ until

$$\frac{\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_\infty}{\|\mathbf{x}^{(k)}\|_\infty} < 10^{-3}.$$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

$$
\begin{aligned}
10x_1 - x_2 + 2x_3 \phantom{{}+3x_4} &= 6, \\
-x_1 + 11x_2 - x_3 + 3x_4 &= 25, \\
2x_1 - x_2 + 10x_3 - x_4 &= -11, \\
3x_2 - x_3 + 8x_4 &= 15
\end{aligned}
$$

$$
\begin{aligned}
10x_1 - \quad x_2 + \ 2x_3 \qquad\quad &= 6, \\
-x_1 + 11x_2 - \quad x_3 + 3x_4 &= 25, \\
2x_1 - \quad x_2 + 10x_3 - \ x_4 &= -11, \\
3x_2 - \quad x_3 + 8x_4 &= 15
\end{aligned}
$$

We first solve equation E_i for x_i, for each $i = 1, 2, 3, 4$, to obtain

$$
\begin{aligned}
x_1 &= \qquad\quad \frac{1}{10}x_2 - \frac{1}{5}x_3 \qquad\quad + \frac{3}{5}, \\
x_2 &= \frac{1}{11}x_1 \qquad\quad + \frac{1}{11}x_3 - \frac{3}{11}x_4 + \frac{25}{11}, \\
x_3 &= -\frac{1}{5}x_1 + \frac{1}{10}x_2 \qquad\quad + \frac{1}{10}x_4 - \frac{11}{10}, \\
x_4 &= \qquad\quad - \frac{3}{8}x_2 + \frac{1}{8}x_3 \qquad\quad + \frac{15}{8}.
\end{aligned}
$$

From the initial approximation $\mathbf{x}^{(0)} = (0, 0, 0, 0)^t$ we have $\mathbf{x}^{(1)}$ given by

$$
\begin{aligned}
x_1^{(1)} &= \qquad\qquad\quad \frac{1}{10}x_2^{(0)} - \frac{1}{5}x_3^{(0)} \qquad\qquad + \frac{3}{5} = 0.6000,\\
x_2^{(1)} &= \frac{1}{11}x_1^{(0)} \qquad\qquad + \frac{1}{11}x_3^{(0)} - \frac{3}{11}x_4^{(0)} + \frac{25}{11} = 2.2727,\\
x_3^{(1)} &= -\frac{1}{5}x_1^{(0)} + \frac{1}{10}x_2^{(0)} \qquad\qquad + \frac{1}{10}x_4^{(0)} - \frac{11}{10} = -1.1000,\\
x_4^{(1)} &= \qquad\qquad - \frac{3}{8}x_2^{(0)} + \frac{1}{8}x_3^{(0)} \qquad\qquad + \frac{15}{8} = 1.8750.
\end{aligned}
$$

k	0	1	2	3	4	5	6	7	8	9	10
$x_1^{(k)}$	0.0000	0.6000	1.0473	0.9326	1.0152	0.9890	1.0032	0.9981	1.0006	0.9997	1.0001
$x_2^{(k)}$	0.0000	2.2727	1.7159	2.053	1.9537	2.0114	1.9922	2.0023	1.9987	2.0004	1.9998
$x_3^{(k)}$	0.0000	−1.1000	−0.8052	−1.0493	−0.9681	−1.0103	−0.9945	−1.0020	−0.9990	−1.0004	−0.9998
$x_4^{(k)}$	0.0000	1.8750	0.8852	1.1309	0.9739	1.0214	0.9944	1.0036	0.9989	1.0006	0.9998

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c},$$

We first solve equation E_i for x_i, for each $i = 1, 2, 3, 4$, to obtain

$$\begin{aligned}
x_1 &= \qquad\qquad \frac{1}{10}x_2 - \frac{1}{5}x_3 \qquad\qquad + \frac{3}{5},\\
x_2 &= \frac{1}{11}x_1 \qquad\qquad + \frac{1}{11}x_3 - \frac{3}{11}x_4 + \frac{25}{11},\\
x_3 &= -\frac{1}{5}x_1 + \frac{1}{10}x_2 \qquad\qquad + \frac{1}{10}x_4 - \frac{11}{10},\\
x_4 &= \qquad\qquad - \frac{3}{8}x_2 + \frac{1}{8}x_3 \qquad\qquad + \frac{15}{8}.
\end{aligned}$$

We first solve equation E_i for x_i, for each $i = 1, 2, 3, 4$, to obtain

$$\begin{aligned}
x_1 &= \qquad\quad \frac{1}{10}x_2 - \frac{1}{5}x_3 \qquad\quad + \frac{3}{5}, \\
x_2 &= \frac{1}{11}x_1 \qquad\quad + \frac{1}{11}x_3 - \frac{3}{11}x_4 + \frac{25}{11}, \\
x_3 &= -\frac{1}{5}x_1 + \frac{1}{10}x_2 \qquad\quad + \frac{1}{10}x_4 - \frac{11}{10}, \\
x_4 &= \qquad\quad - \frac{3}{8}x_2 + \frac{1}{8}x_3 \qquad\quad + \frac{15}{8}.
\end{aligned}$$

Hence we have

$$T = \begin{bmatrix} 0 & \frac{1}{10} & -\frac{1}{5} & 0 \\ \frac{1}{11} & 0 & \frac{1}{11} & -\frac{3}{11} \\ -\frac{1}{5} & \frac{1}{10} & 0 & \frac{1}{10} \\ 0 & -\frac{3}{8} & \frac{1}{8} & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} \frac{3}{5} \\ \frac{25}{11} \\ -\frac{11}{10} \\ \frac{15}{8} \end{bmatrix}.$$

ALGORITHM
7.1

Jacobi Iterative

To solve $A\mathbf{x} = \mathbf{b}$ given an initial approximation $\mathbf{x}^{(0)}$:

INPUT the number of equations and unknowns n; the entries a_{ij}, $1 \le i, j \le n$ of the matrix A; the entries b_i, $1 \le i \le n$ of $\mathbf{b}$; the entries XO_i, $1 \le i \le n$ of $\mathbf{XO} = \mathbf{x}^{(0)}$; tolerance TOL; maximum number of iterations N.

OUTPUT the approximate solution $x_1, \dots, x_n$ or a message that the number of iterations was exceeded.

Step 1 Set $k = 1$.

Step 2 While ($k \le N$) do Steps 3–6.

Step 3 For $i = 1, \dots, n$

$$\text{set } x_i = \frac{1}{a_{ii}}\left[-\sum_{\substack{j=1\\ j\ne i}}^{n}(a_{ij}XO_j) + b_i\right].$$

Step 4 If $||\mathbf{x} - \mathbf{XO}|| < TOL$ then OUTPUT $(x_1, \dots, x_n)$;
(*The procedure was successful.*)
STOP.

Step 5 Set $k = k + 1$.

Step 6 For $i = 1, \dots, n$ set $XO_i = x_i$.

Step 7 OUTPUT ('Maximum number of iterations exceeded');
(*The procedure was successful.*)
STOP.

The Gauss-Seidel Method

$$x_i^{(k)} = \frac{1}{a_{ii}}\left[-\sum_{j=1}^{i-1}(a_{ij}x_j^{(k)}) - \sum_{j=i+1}^{n}(a_{ij}x_j^{(k-1)}) + b_i\right], \tag{7.8}$$

for each $i = 1, 2, \ldots, n$, instead of Eq. (7.5). This modification is called the **Gauss-Seidel iterative technique** and is illustrated in the following example.

Use the Gauss-Seidel iterative technique to find approximate solutions to

$$\begin{aligned}
10x_1 - x_2 + 2x_3 \quad &= 6, \\
-x_1 + 11x_2 - x_3 + 3x_4 &= 25, \\
2x_1 - x_2 + 10x_3 - x_4 &= -11, \\
3x_2 - x_3 + 8x_4 &= 15
\end{aligned}$$

starting with $\mathbf{x} = (0, 0, 0, 0)^t$ and iterating until

$$\frac{\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_\infty}{\|\mathbf{x}^{(k)}\|_\infty} < 10^{-3}.$$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

$$
\begin{aligned}
x_1^{(k)} &= \qquad\qquad \frac{1}{10}x_2^{(k-1)} - \frac{1}{5}x_3^{(k-1)} \qquad\qquad + \frac{3}{5}, \\
x_2^{(k)} &= \frac{1}{11}x_1^{(k)} \qquad\qquad + \frac{1}{11}x_3^{(k-1)} - \frac{3}{11}x_4^{(k-1)} + \frac{25}{11}, \\
x_3^{(k)} &= -\frac{1}{5}x_1^{(k)} + \frac{1}{10}x_2^{(k)} \qquad\qquad + \frac{1}{10}x_4^{(k-1)} - \frac{11}{10}, \\
x_4^{(k)} &= \qquad - \frac{3}{8}x_2^{(k)} + \frac{1}{8}x_3^{(k)} \qquad\qquad + \frac{15}{8}.
\end{aligned}
$$

Table 7.2

k	0	1	2	3	4	5
$x_1^{(k)}$	0.0000	0.6000	1.030	1.0065	1.0009	1.0001
$x_2^{(k)}$	0.0000	2.3272	2.037	2.0036	2.0003	2.0000
$x_3^{(k)}$	0.0000	−0.9873	−1.014	−1.0025	−1.0003	−1.0000
$x_4^{(k)}$	0.0000	0.8789	0.9844	0.9983	0.9999	1.0000

Because

$$\frac{\|\mathbf{x}^{(5)} - \mathbf{x}^{(4)}\|_\infty}{\|\mathbf{x}^{(5)}\|_\infty} = \frac{0.0008}{2.000} = 4 \times 10^{-4},$$

ALGORITHM
7.2

Gauss-Seidel Iterative

To solve $A\mathbf{x} = \mathbf{b}$ given an initial approximation $\mathbf{x}^{(0)}$:

INPUT the number of equations and unknowns n; the entries a_{ij}, $1 \le i, j \le n$ of the matrix A; the entries b_i, $1 \le i \le n$ of $\mathbf{b}$; the entries XO_i, $1 \le i \le n$ of $\mathbf{XO} = \mathbf{x}^{(0)}$; tolerance TOL; maximum number of iterations N.

OUTPUT the approximate solution $x_1, \ldots, x_n$ or a message that the number of iterations was exceeded.

Step 1 Set $k = 1$.

Step 2 While $(k \le N)$ do Steps 3–6.

Step 3 For $i = 1, \ldots, n$

$$\text{set } x_i = \frac{1}{a_{ii}} \left[-\sum_{j=1}^{i-1} a_{ij} x_j - \sum_{j=i+1}^{n} a_{ij} XO_j + b_i \right].$$

Step 4 If $||\mathbf{x} - \mathbf{XO}|| < TOL$ then OUTPUT $(x_1, \ldots, x_n)$;
(*The procedure was successful.*)
STOP.

Step 5 Set $k = k + 1$.

Step 6 For $i = 1, \ldots, n$ set $XO_i = x_i$.

Step 7 OUTPUT ('Maximum number of iterations exceeded');
(*The procedure was successful.*)
STOP. ■

Vector Norms: Scalar measure of the magnitude of a vector

Here are some vector norms for n x 1 vectors $\{x\}$ with typical elements x_i.
Each is in the general form of a p norm defined by the general relationship:

$$\| x \|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

1. Sum of the magnitudes: $$\| x \|_1 = \left(\sum_{i=1}^{n} |x_i| \right)$$

2. Magnitude of largest element: (infinity norm) $$\| x \|_\infty = \max_i |x_i|$$

3. Length or Euclidean norm: $$\| x \|_2 = \left(\sum_{i=1}^{n} |x_i|^2 \right)^{1/2}$$

Cover each and everything discussed in class

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Lecture # 46-48

Power Method

For large values of n, polynomial equations like this one are difficult and time-consuming to solve. Moreover, numerical techniques for approximating roots of polynomial equations of high degree are sensitive to rounding errors. In this section we look at an alternative method for approximating eigenvalues. As presented here, the method can be used only to find the eigenvalue of A that is largest in absolute value—we call this eigenvalue the **dominant eigenvalue** of A. Although this restriction may seem severe, dominant eigenvalues are of primary interest in many physical applications.

Definition of Dominant Eigenvalue and Dominant Eigenvector

Let $\lambda_1, \lambda_2, \ldots,$ and λ_n be the eigenvalues of an $n \times n$ matrix A. λ_1 is called the **dominant eigenvalue** of A if

$$|\lambda_1| > |\lambda_i|, \quad i = 2, \ldots, n.$$

The eigenvectors corresponding to λ_1 are called **dominant eigenvectors** of A.

Not every matrix has a dominant eigenvalue. For instance, the matrix

$$A = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

(with eigenvalues of $\lambda_1 = 1$ and $\lambda_2 = -1$) has no dominant eigenvalue. Similarly, the matrix

$$A = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(with eigenvalues of $\lambda_1 = 2$, $\lambda_2 = 2$, and $\lambda_3 = 1$) has no dominant eigenvalue.

Remember

$$Ax = \lambda x$$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

$$A=\begin{bmatrix}1 & -3 & 2\\ 4 & 4 & -1\\ 6 & 3 & 5\end{bmatrix}$$

Complete six iterations of the power method to approximate a dominant eigenvector of

$$A = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}.$$

Solution We begin with an initial nonzero approximation of

$$\mathbf{x}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

We then obtain the following approximations.

Iteration		*Approximation*
$\mathbf{x}_1 = A\mathbf{x}_0 = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -10 \\ -4 \end{bmatrix}$	$\rightarrow$	$-4\begin{bmatrix} 2.50 \\ 1.00 \end{bmatrix}$
$\mathbf{x}_2 = A\mathbf{x}_1 = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} -10 \\ -4 \end{bmatrix} = \begin{bmatrix} 28 \\ 10 \end{bmatrix}$	$\rightarrow$	$10\begin{bmatrix} 2.80 \\ 1.00 \end{bmatrix}$
$\mathbf{x}_3 = A\mathbf{x}_2 = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} 28 \\ 10 \end{bmatrix} = \begin{bmatrix} -64 \\ -22 \end{bmatrix}$	$\rightarrow$	$-22\begin{bmatrix} 2.91 \\ 1.00 \end{bmatrix}$
$\mathbf{x}_4 = A\mathbf{x}_3 = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} -64 \\ -22 \end{bmatrix} = \begin{bmatrix} 136 \\ 46 \end{bmatrix}$	$\rightarrow$	$46\begin{bmatrix} 2.96 \\ 1.00 \end{bmatrix}$
$\mathbf{x}_5 = A\mathbf{x}_4 = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} 136 \\ 46 \end{bmatrix} = \begin{bmatrix} -280 \\ -94 \end{bmatrix}$	$\rightarrow$	$-94\begin{bmatrix} 2.98 \\ 1.00 \end{bmatrix}$
$\mathbf{x}_6 = A\mathbf{x}_5 = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}\begin{bmatrix} -280 \\ -94 \end{bmatrix} = \begin{bmatrix} 568 \\ 190 \end{bmatrix}$	$\rightarrow$	$190\begin{bmatrix} 2.99 \\ 1.00 \end{bmatrix}$

Illustration The matrix

$$A = \begin{bmatrix} -2 & -3 \\ 6 & 7 \end{bmatrix}$$

Has eigenvalues $\lambda_1 = 4$ and $\lambda_2 = 1$ with corresponding eigenvectors $\mathbf{v}_1 = (1, -2)^t$ and $\mathbf{v}_2 = (1, -1)^t$. If we start with the arbitrary vector $\mathbf{x}_0 = (1, 1)^t$ and multiply by the matrix A we obtain

$$\mathbf{x}_1 = A\mathbf{x}_0 = \begin{bmatrix} -5 \\ 13 \end{bmatrix}, \quad \mathbf{x}_2 = A\mathbf{x}_1 = \begin{bmatrix} -29 \\ 61 \end{bmatrix}, \quad \mathbf{x}_3 = A\mathbf{x}_2 = \begin{bmatrix} -125 \\ 253 \end{bmatrix},$$

$$\mathbf{x}_4 = A\mathbf{x}_3 = \begin{bmatrix} -509 \\ 1021 \end{bmatrix}, \quad \mathbf{x}_5 = A\mathbf{x}_4 = \begin{bmatrix} -2045 \\ 4093 \end{bmatrix}, \quad \mathbf{x}_6 = A\mathbf{x}_5 = \begin{bmatrix} -8189 \\ 16381 \end{bmatrix}.$$

As a consequence, approximations to the dominant eigenvalue $\lambda_1 = 4$ are

$$\lambda_1^{(1)} = \frac{61}{13} = 4.6923, \quad \lambda_1^{(2)} = \frac{253}{61} = 4.14754, \quad \lambda_1^{(3)} = \frac{1021}{253} = 4.03557,$$

$$\lambda_1^{(4)} = \frac{4093}{1021} = 4.00881, \quad \lambda_1^{(5)} = \frac{16381}{4093} = 4.00200.$$

An approximate eigenvector corresponding to $\lambda_1^{(5)} = \dfrac{16381}{4093} = 4.00200$ is

$$\mathbf{x}_6 = \begin{bmatrix} -8189 \\ 16381 \end{bmatrix}, \quad \text{which, divided by 16381, normalizes to} \quad \begin{bmatrix} -0.49908 \\ 1 \end{bmatrix} \approx \mathbf{v}_1.$$

□

Do Q1 & 2 from Exercise

National University
of computer and emerging sciences

Foundation for Advancement of Science and Technology

Final Exam Syllabus (for students of all NC sections)

from
Numerical Analysis
NINTH EDITION
Richard L. Burden
J. Douglas Faires

1-Error Analysis

Introduction of Numerical Computing ,
Chopping, Roundoff and truncation error ,
Absolute, relative and percentage error.
Taylor polynomial, Significant figures,

2-Interpolation and Polynomial approximation

Topic 3.1 page no (124-129)
Lagrange interpolation polynomial of degree one, two and three.
Divided difference table and interpolating polynomial.
Newton Forward and Backward difference formula
Questions 1,2,5,6,13,14,19
Topic 3.3: Stirling Formula (Center difference Formula)

2- Numerical Differentiation

Topic 4.1: All the concepts covered in questions
1,2,5,6,18 and 26

3- Numerical Integration

Topic 4.3: All the concepts covered in questions 1-10
Topic 4.4: All the concepts covered in questions 1-4

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

4- Solving ODEs

Topic 5.2: (Euler's Method) All topics covered in question no 1-4.
RK-2 Method
Special cases of RK-2 Method Heun's and Midpoint method
RK-4 Method
All topics covered in ODE practice sheet.

5-Direct Method for solving linear system

Topic 6.5: LU decomposition (Dolittle)
Topic 6.6: Positive definite matrices, LDL^t Factorization , Crout and Cholesky method

6-Iterative Method for solving linear system

Topic 7.3: Gauss-Siedel and Jacobi's methods with L-1, L-2 and L-infinity Norm.

7-Approximating Eigen values and Eigen vectors

Topic 9.3: Power Method

8-Numerical Optimization:

Gradient Descent Algorithm (Assignment)

9-Lab Sessions

Lab Session2
Lab Session 3a
Lab Session 3b
Lab Session 4

Gradient Descent

x	y
0.23993	2.81443
0.80287	3.30876
0.10465	0.41829
0.30075	-0.37122
0.53666	2.22992
0.08107	1.76152
0.18509	2.84154
0.96605	1.15965
0.32419	0.29688
0.38713	-0.13069

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

$$h_\theta(x) = \sum_{j=0}^{1} \theta_j x_j$$

$$h_{\theta_0,\theta_1}(x) = \theta_0 x_0 + \theta_1 x_1$$

where $x_0 = 1$

$$\therefore \boxed{h_{\theta_0,\theta_1}(x) = \theta_0 + \theta_1 x_1}$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_{\theta_0,\theta_1}(x_{1i}) - y_i\right)^2$$

$$m = 10$$

$$J(\theta_0, \theta_1) = \frac{1}{2(10)} \Big[(\theta_0 + 0.23993\theta_1 - 2.81443)^2 + (\theta_0 + 0.80287\theta_1 - 3.30876)^2 + (\theta_0 + 0.10465\theta_1 - 0.41829)^2 + (\theta_0 + 0.30075\theta_1 + 0.37122)^2 + (\theta_0 + 0.53666\theta_1 - 2.22992)^2 + (\theta_0 + 0.08107\theta_1 - 1.76152)^2 + (\theta_0 + 0.18509\theta_1 - 2.84154)^2 + (\theta_0 + 0.96605\theta_1 - 1.15965)^2 + (\theta_0 + 0.32419\theta_1 - 0.29688)^2 + (\theta_0 + 0.38713\theta_1 + 0.13069)^2 \Big]$$

$$\boxed{J(\theta_0, \theta_1) = \frac{1}{20} \left(10\theta_0^2 + 7.856780\theta_0\theta_1 + 2.32062\theta_1^2 - 28.65816\theta_0 - 12.59057\theta_1 + 36.78153\right)}$$

$$\frac{\partial J}{\partial \theta_0} = \cancel{2} \times \frac{1}{\cancel{2}m} \sum_{i=1}^{m} \left(h_{\theta_0,\theta_1}(x_{1i}) - y_i\right) \times \frac{\partial}{\partial \theta_0}\left(\theta_0 + \theta_1 x_{1i} - y_i\right)$$

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{m} \sum_{i=1}^{m} \left(h_{\theta_0,\theta_1}(x_{1i}) - y_i\right)$$

$$\begin{aligned}
\frac{\partial J}{\partial \theta_0} = \frac{1}{10} \Big[& (\theta_0 + 0.23993\theta_1 - 2.81443) + \\
& (\theta_0 + 0.80287\theta_1 - 3.30876) + \\
& (\theta_0 + 0.10465\theta_1 - 0.41829) + \\
& (\theta_0 + 0.30075\theta_1 + 0.37122) + \\
& (\theta_0 + 0.53666\theta_1 - 2.22992) + \\
& (\theta_0 + 0.08107\theta_1 - 1.76152) + \\
& (\theta_0 + 0.18509\theta_1 - 2.84154) + \\
& (\theta_0 + 0.96605\theta_1 - 1.15965) + \\
& (\theta_0 + 0.32419\theta_1 - 0.29688) + \\
& (\theta_0 + 0.38713\theta_1 + 0.13069) \Big]
\end{aligned}$$

$$\boxed{\frac{\partial J}{\partial \theta_0} = \frac{1}{10}\left(10\theta_0 + 3.92839\theta_1 - 14.32908\right)}$$

National University of computer and emerging sciences

Foundation for Advancement of Science & Technology

$$\frac{\partial J}{\partial \theta_1} = \cancel{2} \times \frac{1}{\cancel{2}m} \sum_{i=1}^{m} (h_{\theta_0,\theta_1}(x_{1i}) - y_i) \times \frac{\partial}{\partial \theta_1}(\theta_0 + \theta_1 x_{1i} - y_i)$$

$$\frac{\partial J}{\partial \theta_1} = \frac{1}{m} \sum_{i=1}^{m} (x_{1i}(h_{\theta_0,\theta_1}(x_{1i}) - y_i))$$

$$\frac{\partial J}{\partial \theta_1} = \frac{1}{10}\Big[(0.23993(\theta_0 + 0.23993\theta_1 - 2.81443)) + (0.80287(\theta_0 + 0.80287\theta_1 - 3.30876)) + (0.10465(\theta_0 + 0.10465\theta_1 - 0.41829)) + (0.30075(\theta_0 + 0.30075\theta_1 + 0.37122)) + (0.53666(\theta_0 + 0.53666\theta_1 - 2.22992)) + (0.08107(\theta_0 + 0.08107\theta_1 - 1.76152)) + (0.18509(\theta_0 + 0.18509\theta_1 - 2.84154)) + (0.96605(\theta_0 + 0.96605\theta_1 - 1.15965)) + (0.32419(\theta_0 + 0.32419\theta_1 - 0.29688)) + (0.38713(\theta_0 + 0.38713\theta_1 + 0.13069))\Big]$$

$$\boxed{\frac{\partial J}{\partial \theta_1} = \frac{1}{10}(3.92839\,\theta_0 + 2.32062\theta_1 - 6.29528)}$$

ITERATIONS :-

$$J(\theta_0,\theta_1) = \frac{1}{20}(10\theta_0^2 + 7.85678\theta_0\theta_1 + 2.32062\theta_1^2 - 28.65816\theta_0 - 12.59057\theta_1 + 36.78153)$$

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{10}(10\theta_0 + 3.92839\theta_1 - 14.32908)$$

$$\frac{\partial J}{\partial \theta_1} = \frac{1}{10}(3.92839\theta_0 + 2.32062\theta_1 - 6.29528)$$

ITERATION 0 :

Initially,

$\theta_0 = 0$, $\theta_1 = 0$

$$J(0,0) = \frac{1}{20}\left[10(0)^2 + 7.85678(0)(0) + 2.32062(0)^2 - 28.65816(0) - 12.59057(0) + 36.78153\right]$$

$$J(0,0) = \frac{1}{20}(36.78153)$$

$$J(0,0) = 1.83907$$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{10}\left[10(0) + 3.92839(0) - 14.32908\right]$$

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{10}(-14.32908)$$

$$\frac{\partial J}{\partial \theta_0} = -1.432908$$

$$\frac{\partial J}{\partial \theta_1} = \frac{1}{10}\left[3.92839(0) + 2.32062(0) - 6.29528\right]$$

$$\frac{\partial J}{\partial \theta_1} = \frac{1}{10}(-6.29528)$$

$$\frac{\partial J}{\partial \theta_1} = -0.62952$$

New $\theta_0 = \theta_0 - \alpha \times \frac{\partial J}{\partial \theta_0}$

$$\theta_0 = 0 - 0.5 \times (-1.432908)$$

$$\theta_0 = 0.71645$$

New $\theta_1 = \theta_1 - \alpha \times \frac{\partial J}{\partial \theta_1}$

$$\theta_1 = 0 - 0.5 \times (-0.62952)$$

$$\theta_1 = 0.31476$$

ITERATION 1:

$\theta_0 = 0.71645$, $\theta_1 = 0.31476$

$$J(0.71645, 0.31476) = 0.97105$$

$$\frac{\partial J}{\partial \theta_0} = -0.59280$$

$$\frac{\partial J}{\partial \theta_1} = -0.27503$$

$$\theta_0 = \theta_0 - \alpha \times \frac{\partial J}{\partial \theta_0}$$

$$\theta_0 = 0.71645 - 0.5 \times (-0.59280)$$

$$\theta_0 = 1.01285$$

$$\theta_1 = \theta_1 - \alpha \times \frac{\partial J}{\partial \theta_1}$$

$$\theta_1 = 0.31476 - 0.5 \times (-0.27503)$$

$$\theta_1 = 0.45228$$

ITERATION 2:

$\theta_0 = 1.01285$, $\theta_1 = 0.45228$

$$J(1.01285, 0.45228) = 0.81965$$

$$\frac{\partial J}{\partial \theta_0} = -0.24238$$

$$\frac{\partial J}{\partial \theta_1} = -0.12668$$

$$\theta_0 = \theta_0 - \alpha \times \frac{\partial J}{\partial \theta_0}$$

$$\theta_0 = 1.01285 - 0.5 \times (-0.24238)$$

$$\theta_0 = 1.13404$$

$$\theta_1 = \theta_1 - \alpha \times \frac{\partial J}{\partial \theta_1}$$

$$\theta_1 = 0.45228 - 0.5 \times (-0.12668)$$

$$\theta_1 = 0.51562$$

ITERATION 3:

$\theta_0 = 1.13404$, $\theta_1 = 0.51562$

$$J(1.13404, 0.51562) = 0.79308$$

$$\frac{\partial J}{\partial \theta_0} = -0.09631$$

$$\frac{\partial J}{\partial \theta_1} = -0.06437$$

$$\theta_0 = \theta_0 - \alpha \times \frac{\partial J}{\partial \theta_0}$$

$$\theta_0 = 1.13404 - 0.5 \times (-0.09631)$$

$$\theta_0 = 1.18219$$

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

$\theta_1 = \theta_1 - \alpha \times \frac{\partial J}{\partial \theta_1}$

$\theta_1 = 0.51562 - 0.5 \times (-0.06437)$

$\theta_1 = 0.54780$

ITERATION 4:

$\theta_0 = 1.18219$; $\theta_1 = 0.54780$

$\boxed{J(1.18219, 0.54780) = 0.78826}$

TABLE :-

Iteration No.	θ_0	θ_1	$J(\theta_0, \theta_1)$
0	0	0	1.83907
1	0.71645	0.31476	0.97105
2	1.01285	0.45228	0.81965
3	1.13404	0.51562	0.79308
4	1.18219	0.54780	0.78826

Practice according to what we have discussed in our class