

www.vuCTn.com

MTH603 MERGED

PPT Slides 1 TO 45

Regards : Zarva Chaudhary

Merged File date 22/12/2022

Admin:

* Zarva Chaudhary *

* Chandhary Moazzam *

* Laiba Maki *

Numerical Analysis Lecture 1

Numerical

Computation

Numerical Methods

Numerical Analysis

Rs. 2.1234567

2.12

Rs 2.1234567 Th

2,123.45

Rs 2.1234567 M

2,123,456.70

Millennium Bug

2000

September 24, 1978

092478

092498

092499

123199

123199

010110

010120

010100

12311999

01012000

Bank Account

Profit as %age

602.55565 Rupees

0.00565

x 100

0.565

x 1,000

5.65

x 1,000,000

5,650.00

Course Contents

Introduction

**Solution of Non Linear
Equations**

Solution of Linear

System of Equations

**Approximation of Eigen
Values**

**Interpolation and
Polynomial Approximation
Numerical Differentiation
Numerical Integration
Numerical Solution of
Ordinary Differential
Equations**

Chapter 1

Introduction

Number (s)

System (s)

In our daily life, we use numbers based on the decimal system.

In this system, we use ten symbols 0, 1, ..., 9 and the number 10 is called the base of the system.

**Thus, when a base
 N is given, we
need N different
symbols**

**$0, 1, 2, \dots, (N - 1)$
to represent an
arbitrary number.**

The number systems commonly used in computers are

Base, N	Number
2	Binary
8	Octal
10	Decimal
16	Hexadecimal

An arbitrary real number, a can be written as

$$\begin{aligned} a = & a_m N^m + a_{m-1} N^{m-1} + \dots + \\ & a_1 N^1 + a_0 + a_{-1} N^{-1} \\ & + \dots + a_{-m} N^{-m} \end{aligned}$$

In binary system, it has the form,

$$a = a_m 2^m + a_{m-1} 2^{m-1} + \cdots + a_1 2^1 + a_0 + a_{-1} 2^{-1} + \cdots + a_{-m} 2^{-m}$$

The decimal number 1729 is represented and calculated

$$(1729)_{10} = 1 \times 10^3 + 7 \times 10^2 + 2 \times 10^1 + 9 \times 10^0$$

While the decimal equivalent of binary number 10011001 is

$$1 \times 2^0 + 0 \times 2^{-1} + 0 \times 2^{-2}$$

$$+ 1 \times 2^{-3} + 1 \times 2^{-4} + 0 \times 2^{-5}$$

$$+ 0 \times 2^{-6} + 1 \times 2^{-7}$$

$$= 1 + \frac{1}{8} + \frac{1}{16} + \frac{1}{128}$$

$$= (1.1953125)_{10}$$

Electronic computers use binary system whose base is 2. The two symbols used in this system are 0 and 1, which are called *binary digits* or simply *bits*.

The internal representation of any data within a computer is in binary form. However, we prefer data input and output of numerical results in decimal system. Within the computer, the arithmetic is carried out in binary form.

Conversion of decimal number 47 into its binary equivalent

2	47	Remainder
2	23	1
2	11	1
2	5	1
2	2	1
2	1	0
	0	1 Most significant bit

$$(47)_{10} = (101111)_2$$

Binary equivalent of the decimal fraction 0.7625.

		Product	Integer
0.7625	x2	1.5250	1
0.5250	x2	1.0500	1
0.05	x2	0.1	0
0.1	x2	0.2	0
0.2	x2	0.4	0
0.4	x2	0.8	0
0.8	x2	1.6	1
0.6	x2	1.2	1
0.2	x2	0.4	0

$$(0.7625)_{10} = (0.11\dots11(0011))_2$$

Conversion $(59)_{10}$ into binary and then into octal.

2	59	
2	29	1
2	14	1
2	7	0
2	3	1
2	1	1
	0	1

$$(59)_{10} = (11011)_2$$

$$(111011)_2 = 111011 = (73)_8$$

Numerical Analysis Lecture 1

Numerical Analysis Lecture 2

Introduction

**Solution of Non Linear
Equations**

**Solution of Linear System
of Equations**

**Approximation of Eigen
Values**

**Interpolation and
Polynomial Approximation
Numerical Differentiation
Numerical Integration
Numerical Solution of
Ordinary Differential
Equations**

Errors in Computations

Numerically, computed solutions are subject to certain errors. It may be fruitful to identify the error sources and their growth while classifying the errors in numerical computation.

These are
Inherent errors,
local round-off errors
local truncation errors

Inherent errors

It is that quantity of error which is present in the statement of the problem itself, before finding its solution. It arises due to the simplified assumptions made in the mathematical modeling of a problem.

It can also arise when the data is obtained from certain physical measurements of the parameters of the problem.

Local round-off errors

Every computer has a finite word length and therefore it is possible to store only a fixed number of digits of a given input number.

Since computers store information in binary form, storing an exact decimal number in its binary form into the computer memory gives an error. This error is computer dependent.

At the end of computation of a particular problem, the final results in the computer, which is obviously in binary form, should be converted into decimal form-a form understandable to the user-before their print out. Therefore, an additional error is committed at this stage too.

This error is called *local round-off error*.

$$(0.7625)_{10} = (0.11000011(0011))_2$$

If a particular computer system has a word length of 12 bits only, then the decimal number 0.7625 is stored in the computer memory in binary form as 0.110000110011. However, it is equivalent to 0.76245.

Thus, in storing the number 0.7625, we have committed an error equal to 0.00005, which is the round-off error; inherent with the computer system considered.

Thus, we define the *error* as

$$\text{Error} = \text{True value} - \text{Computed value}$$

***Absolute* error, denoted by |Error|, while, the *relative error* is defined as**

Relative error
$$= \frac{|\text{Error}|}{|\text{True value}|}$$

Local truncation error

It is generally easier to expand a function into a power series using Taylor series expansion and evaluate it by retaining the first few terms. For example, we may approximate the function $f(x) = \cos x$ by the series

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots + (-1)^n \frac{x^{2n}}{(2n)!} + \cdots$$

If we use only the first three terms to compute $\cos x$ for a given x , we get an approximate answer. Here, the error is due to truncating the series. Suppose, we retain the first n terms, the truncation error (TE) is given by

$$\text{TE} \leq \frac{x^{2n+2}}{(2n+2)!}$$

The TE is independent of the computer used.

If we wish to compute $\cos x$ for accurate with five significant digits, the question is, how many terms in the expansion are to be included? In this situation

$$\frac{x^{2n+2}}{(2n+2)!} < .5 \times 10^{-5} = 5 \times 10^{-6}$$

Taking logarithm on both sides, we get

$$(2n + 2) \log x - \log[(2n + 2)!]$$

$$< \log_{10} 5 - 6 \log_{10} 10 = 0.699 - 6 = -5.3$$

or $\log[(2n + 2)!] - (2n + 2) \log x > 5.3$

We can observe that, the above inequality is satisfied for $n = 7$. Hence, seven terms in the expansion are required to get the value of $\cos x$, with the prescribed accuracy

The truncation error is given by

$$\text{TE} \leq \frac{x^{16}}{16!}$$

Numerical Analysis Lecture 2

Numerical Analysis Lecture 3

Chapter 2

Solution of Non-Linear Equations

Introduction

Bisection Method

Regula-Falsi Method

Method of iteration

Newton - Raphson Method

Muller's Method

Graeffe's Root Squaring

Method

Definition: The equation $f(x) = 0$ is called an algebraic, if it is purely a polynomial in x ;

it is a transcendental if $f(x)$ contains trigonometric, exponential or logarithmic functions

For example,

$$x^3 + 5x^2 - 6x + 3 = 0$$

is an algebraic equation,

Whereas $M = E - e \sin E$

and $ax^2 + \log(x - 3) + e^x \sin x = 0$

are transcendental equations.

To find the solution of an equation $f(x) = 0$, we find those values of x for which $f(x) = 0$ is satisfied.

Such values of x are called the roots of $f(x) = 0$.

Thus a is a root of an equation $f(x) = 0$, if and only if, $f(a) = 0$.

Properties of an algebraic equation

Every algebraic equation of *n-th* degree, where *n* is a positive integer, has *n* and only *n* roots.

Complex roots occur in pairs.
That is, if $(a + ib)$ is a root of $f(x) = 0$, then $(a - ib)$ is also a root of this equation.

If $x = a$ is a root of $f(x) = 0$, a polynomial of degree n , then $(x - a)$ is a factor of $f(x)$. On dividing $f(x)$ by $(x - a)$ we obtain a polynomial of degree $(n - 1)$.

Descartes rule of signs:

The number of positive roots of an algebraic equation $f(x) = 0$ with real coefficients cannot exceed the number of changes in sign of the coefficients in the polynomial $f(x) = 0$. Similarly, the number of negative roots of $f(x) = 0$ cannot exceed the number of changes in the sign of the coefficients of $f(-x) = 0$.

For example, consider an equation

$$x^3 - 3x^2 + 4x - 5 = 0$$

As there are three changes in sign, so, the degree of the equation is three, and hence the given equation will have all the three positive roots.

Intermediate value property:

If $f(x)$ is a real valued continuous function in the closed interval

$$a \leq x \leq b$$

**If $f(a)$ and $f(b)$ have opposite
signs; that is $f(x) = 0$ has at least
one root β such that**

$$a < \beta < b$$

Numerical methods for solving either a transcendental equation or an algebraic equation are classified into two groups:

Direct methods, which require no knowledge of the initial approximation of a root of the equation $f(x) = 0$.

Iterative methods, require first approximation to initiate iteration.

How to get the first approximation?

We can find the approximate value of the root of $f(x) = 0$ either by graphical method or by an analytical method

Graphical method

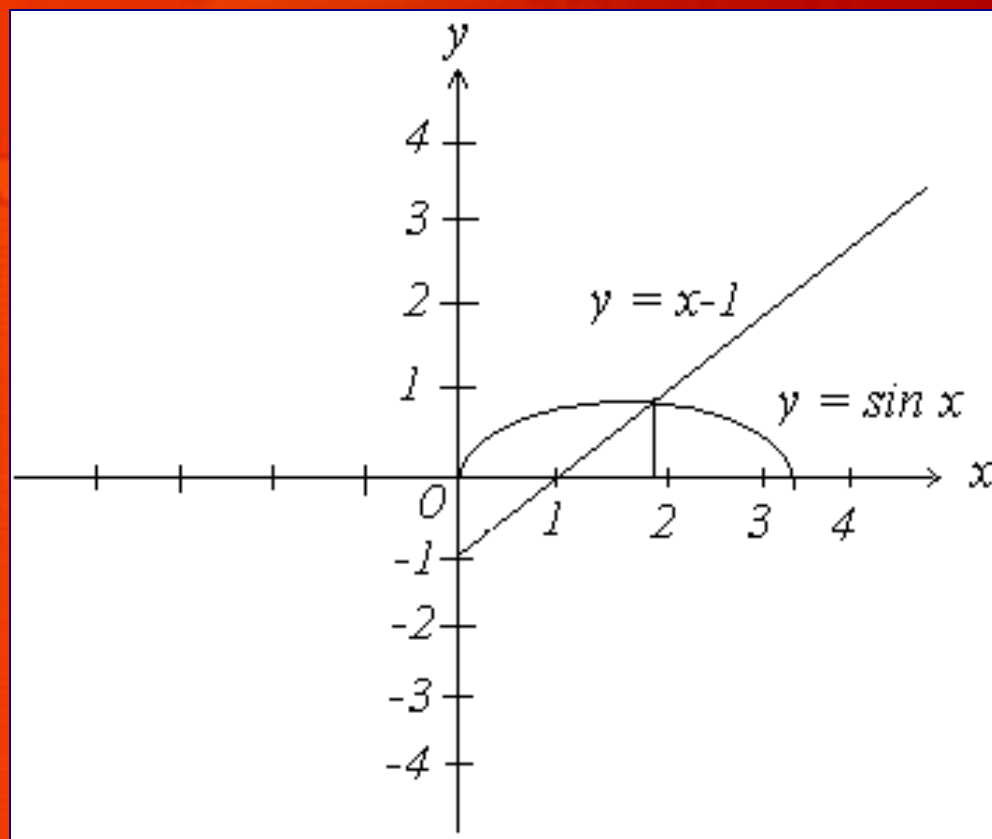
The equation $f(x) = 0$ can be rewritten as $f_1(x) = f_2(x)$ and the first approximation to a root of $f(x) = 0$ can be taken as the abscissa of the point of intersection of the graphs of $y = f_1(x)$ and $y = f_2(x)$.

For example, consider,

$$f(x) = x - \sin x - 1 = 0$$

It can be written as $x - 1 = \sin x$.

Now, we shall draw the graphs of $y = (x - 1)$ and $y = \sin x$



Answer !

**The approximate
value of the root
is found to be 1.9**

Analytical method

This method is based on
'intermediate value property'.

$$f(x) = 3x - \sqrt{1 + \sin x} = 0$$

$$f(0) = -1$$

$$f(1) = 3 - \sqrt{1 + \sin\left(1 \times \frac{180}{\pi}\right)} = 3 - \sqrt{1 + 0.84147} = 1.64299$$

Here $f(0)$ and $f(1)$ are of opposite signs. Therefore, using intermediate value property we infer that there is at least one root between $x = 0$ and $x = 1$. This method is often used to find the first approximation to a root of either transcendental equation or algebraic equation.

Hence, in analytical method, we must always start with an initial interval (a, b) , so that $f(a)$ and $f(b)$ have opposite signs.

Bisection Method (Bolzano)

Suppose, we wish to locate the root of an equation $f(x) = 0$ in an interval, say (x_0, x_1) . Let $f(x_0)$ and $f(x_1)$ are of opposite signs, such that $f(x_0) f(x_1) < 0$.

Then the graph of the function crosses the x -axis between x_0 and x_1 , which guarantees the existence of at least one root in the interval (x_0, x_1) .

The desired root is approximately defined by the midpoint

$$x_2 = \frac{x_0 + x_1}{2}$$

If $f(x_2) = 0$, then x_2 is the desired root of $f(x) = 0$.

However, if $f(x_2) \neq 0$ then the root may be between x_0 and x_2 or x_2 and x_1 .

Now, we define the next approximation by

$$x_3 = \frac{x_0 + x_2}{2}$$

provided $f(x_0) f(x_2) < 0$, then the root may be found between x_0 and x_2 or by

$$x_3 = \frac{x_1 + x_2}{2}$$

provided $f(x_1) f(x_2) < 0$, then the root lies between x_1 and x_2 etc.

Thus, at each step, we either find the desired root to the required accuracy or narrow the range to half the previous interval.

This process of halving the intervals is continued to determine a smaller and smaller interval within which the desired root lies.

Continuation of this process eventually gives us the desired root.

Example

**Solve $x^3 - 9x + 1 = 0$
for the root between
 $x = 2$ and $x = 4$ by the
bisection method.**

Solution Given $f(x) = x^3 - 9x + 1$.

Here $f(2) = -9$, $f(4) = 29$.

Therefore, $f(2) f(4) < 0$ and hence the root lies between 2 and 4.

Let $x_0 = 2$, $x_1 = 4$. Now, we define

$$x_2 = \frac{x_0 + x_1}{2} = \frac{2 + 4}{2} = 3$$

as a first approximation to a root of $f(x) = 0$ and note that $f(3) = 1$, so that $f(2) f(3) < 0$. ***Thus the root lies between 2 and 3***

We further define,

$$x_3 = \frac{x_0 + x_2}{2} = \frac{2 + 3}{2} = 2.5$$

and note that $f(x_3) = f(2.5) < 0$, so that $f(2.5) f(3) < 0$. Therefore, we define the mid-point,

$$x_4 = \frac{x_3 + x_2}{2} = \frac{2.5 + 3}{2} = 2.75, \text{ etc.}$$

Similarly, $x_5 = 2.875$ and $x_6 = 2.9375$

and the process can be continued until the root is obtained to the desired accuracy.

These results are presented in the table.

n	x_n	$f(x_n)$
2	3	1.0
3	2.5	-5.875
4	2.75	-2.9531
5	2.875	-1.1113
6	2.9375	-0.0901

Numerical Analysis Lecture 3

Numerical Analysis Lecture 4

Chapter 2

Solution of Non-Linear Equations

Introduction

Bisection Method

Regula-Falsi Method

Method of iteration

Newton - Raphson Method

Muller's Method

Graeffe's Root Squaring

Method

Bisection Method (Bolzano)

Example

**Solve $x^3 - 9x + 1 = 0$
for the root between
 $x = 2$ and $x = 4$ by the
bisection method.**

Solution Given $f(x) = x^3 - 9x + 1$.

Here $f(2) = -9$, $f(4) = 29$.

Therefore, $f(2) f(4) < 0$ and hence the root lies between 2 and 4.

Let $x_0 = 2$, $x_1 = 4$. Now, we define

$$x_2 = \frac{x_0 + x_1}{2} = \frac{2 + 4}{2} = 3$$

as a first approximation to a root of $f(x) = 0$ and note that $f(3) = 1$, so that $f(2) f(3) < 0$. ***Thus the root lies between 2 and 3***

We further define,

$$x_3 = \frac{x_0 + x_2}{2} = \frac{2 + 3}{2} = 2.5$$

and note that $f(x_3) = f(2.5) < 0$, so that $f(2.5) f(3) < 0$. Therefore, we define the mid-point,

$$x_4 = \frac{x_3 + x_2}{2} = \frac{2.5 + 3}{2} = 2.75, \text{ etc.}$$

Similarly, $x_5 = 2.875$ and $x_6 = 2.9375$

and the process can be continued until the root is obtained to the desired accuracy.

These results are presented in the table.

n	x_n	$f(x_n)$
2	3	1.0
3	2.5	-5.875
4	2.75	-2.9531
5	2.875	-1.1113
6	2.9375	-0.0901

Regula-Falsi Method

Method of false
position

Here, we choose two points x_n and x_{n-1} such that $f(x_n)$ and $f(x_{n-1})$ are of opposite signs. Intermediate value property suggests that the graph of $y = f(x)$ crosses the x -axis between these two points and therefore, a root say α lies between these two points.

Thus, to find a real root of $f(x) = 0$ using Regula-Falsi method, we replace the part of the curve between the points $A[x_n, f(x_n)]$ and $B[x_{n-1}, f(x_{n-1})]$ by a chord in that interval and we take the point of intersection of this chord with the x -axis as a first approximation to the root.

Now, the equation of the chord joining the points A and B is

$$\frac{y - f(x_n)}{f(x_{n-1}) - f(x_n)} = \frac{x - x_n}{x_{n-1} - x_n} \quad (2.1)$$

Setting $y = 0$ in Eq. (2.1), we get

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

Hence, the first approximation to the root of $f(x) = 0$ is given by

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n) \quad (2.2)$$

we observe that $f(x_{n-1})$ and $f(x_{n+1})$ are of opposite sign. Thus, it is possible to apply to above procedure, to determine the line through B and A_1 and so on. Hence, the successive approximations to the root of $f(x) = 0$ is given by Eq. (2.2).

Example

Use the Regula-Falsi method to compute a real root of the equation $x^3 - 9x + 1 = 0$,

- (i) if the root lies between 2 and 4
- (ii) if the root lies between 2 and 3.

Comment on the results.

Solution

Let $f(x) = x^3 - 9x + 1$.

$f(2) = -9$ and $f(4) = 29$.

Since $f(2)$ and $f(4)$ are of opposite signs, the root of $f(x) = 0$ lies between 2 and 4.

Taking $x_1 = 2$, $x_2 = 4$ and using Regula-Falsi method, the first approximation is given by

$$x_3 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2)$$
$$= 4 - \frac{2 \times 29}{38} = 2.47368$$

Now $f(x_3) = -6.12644$.

Since $f(x_2)$ and $f(x_3)$ are of opposite signs, the root lies between x_2 and x_3 .

The second approximation to the root is given as

$$\begin{aligned}x_4 &= x_3 - \frac{x_3 - x_2}{f(x_3) - f(x_2)} f(x_3) \\ &= 2.73989\end{aligned}$$

Therefore $f(x_4) = -3.090707$.

Now, since $f(x_2)$ and $f(x_4)$ are of opposite signs, the third approximation is obtained from

$$x_5 = x_4 - \frac{x_4 - x_2}{f(x_4) - f(x_2)} f(x_4) = 2.86125$$

and $f(x_5) = -1.32686$.

This procedure can be continued till we get the desired result. The first three iterations are shown as in the table.

n	x_{n+1}	$f(x_{n+1})$
2	2.47368	-6.12644
3	2.73989	-3.090707
4	2.86125	-1.32686

(ii) $f(2) = -9$ and $f(3) = 1$. Since $f(2)$ and $f(3)$ are of opposite signs, the root of $f(x) = 0$ lies between 2 and 3. Taking $x_1 = 2$, $x_2 = 3$ and using Regula-Falsi method, the first approximation is given by

$$x_3 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2)$$

$$= 3 - \frac{1}{10} = 2.9$$

$$f(x_3) = -0.711$$

Since $f(x_2)$ and $f(x_3)$ are of opposite signs, the root lies between x_2 and x_3 .

The second approximation to the root is given as

$$x_4 = x_3 - \frac{x_3 - x_2}{f(x_3) - f(x_2)} f(x_3) = 2.94156$$

$$f(x_4) = -0.0207$$

Now, we observe that $f(x_2)$ and $f(x_4)$ are of opposite signs, the third approximation is obtained from

$$\begin{aligned}x_5 &= x_4 - \frac{x_4 - x_2}{f(x_4) - f(x_2)} f(x_4) \\ &= 2.94275\end{aligned}$$

$$f(x) = -0.0011896$$

This procedure can be continued till we get the desired result. The first three iterations are shown as in the table.

n	x_{n+1}	$f(x_{n+1})$
2	2.9	-0.711
3	2.94156	-0.0207
4	2.94275	-0.0011896

We observe that the value of the root as a third approximation is evidently different in both the cases, while the value of x_5 , when the interval considered is $(2, 3)$, is closer to the root.

Important observation: The initial interval (x_1, x_2) in which the root of the equation lies should be sufficiently small.

Example

Use Regula-Falsi method to find a real root of the equation

$$\log x - \cos x = 0$$

accurate to four decimal places after three successive approximations.

Solution

Given $f(x) = \log x - \cos x$.

We observe that

$$f(1) = 0 - 0.5403, \text{ and}$$

$$f(2) = 0.69315 + 0.41615 = 1.1093$$

Since $f(1)$ and $f(2)$ are of opposite signs, the root lies between $x_1 = 1$, $x_2 = 2$.

The first approximation is obtained from

$$x_3 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2)$$

$$= 2 - \frac{1.1093}{1.6496} = 1.3275$$

$$f(x_3) = 0.2833 - 0.2409 = 0.0424$$

Now, since $f(x_1)$ and $f(x_3)$ are of opposite signs, the second approximation is obtained as

$$x_4 = 1.3275 - \frac{(.3275)(.0424)}{0.0424 + 0.5403}$$
$$= 1.3037$$

$$f(x_4) = 1.24816 \times 10^{-3}$$

Similarly, we observe that $f(x_1)$ and $f(x_4)$ are of opposite signs, so, the third approximation is given by

$$x_5 = 1.3037 - \frac{(0.3037)(0.001248)}{0.001248 + 0.5403} \\ = 1.3030$$

$$f(x_5) = 0.6245 \times 10^{-4}$$

The required real root is 1.3030.

Numerical Analysis Lecture 4

Numerical Analysis Lecture 5

Chapter 2

Solution of Non-Linear Equations

Introduction

Bisection Method

Regula-Falsi Method

Method of iteration

Newton - Raphson Method

Muller's Method

Graeffe's Root Squaring

Method

Bisection Method (Bolzano)

Regula-Falsi Method

**Method of false
position**

Here, we choose two points x_n and x_{n-1} such that $f(x_n)$ and $f(x_{n-1})$ are of opposite signs. Intermediate value property suggests that the graph of $y = f(x)$ crosses the x -axis between these two points and therefore, the root lies between these two points.

Example

Using Regula-Falsi method,
find the real root of the
following equation correct,
to three decimal places:

$$x \log_{10} x = 1.2$$

Solution

Let $f(x) = x \log_{10} x - 1.2$

$$f(2) = -0.5979,$$

$$f(3) = 0.2314.$$

Since $f(2)$ and $f(3)$ are of opposite signs, the real root lies between

$$x_1 = 2, x_2 = 3.$$

The first approximation is obtained from

$$x_3 = x_2 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_2)$$

$$= 3 - \frac{0.2314}{0.8293} = 2.72097$$

$$f(x_3) = -0.01713.$$

Since $f(x_2)$ and $f(x_3)$ are of opposite signs, the root of $f(x) = 0$ lies between x_2 and x_3 . Now, the second approximation is given by

$$x_4 = x_3 - \frac{x_3 - x_2}{f(x_3) - f(x_2)} f(x_3) = 2.7402$$

$$f(x_4) = -3.8905 \times 10^{-4}$$

Answer !

**Thus, the root of the
given equation
correct to three
decimal places is
2.740**

Method of Iteration

METHOD OF ITERATION can be applied to find a real root of the equation $f(x) = 0$ by rewriting the same in the form,

$$x = \phi(x) \quad (2.3)$$

Example,

$$f(x) = \cos x - 2x + 3 = 0.$$

It can be

rewritten as

$$x = \frac{1}{2}(\cos x + 3) = \phi(x)$$

Also, $f'(x)$ and $f''(x)$ do not vanish in $(0, 1)$ and $f(x)$ and $f''(x)$ will have the same sign at $x = 1$.

Therefore, we take the first approximation $x_0 = 1$, and using N-R method, we get

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = \frac{e+2}{2e} = 0.867879$$

$$f(x_1) = 6.71607 \times 10^{-2}$$

The second approximation is

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 0.867879 - \frac{0.06716}{4.44902} = 0.85278$$

$$f(x_2) = 7.655 \times 10^{-4}$$

The required root is 0.853.

Example

Find a real root of the equation $x^3 - x - 1 = 0$ using Newton - Raphson method, correct to four decimal places.

Numerical Analysis Lecture 5

Numerical Analysis Lecture 6

Chapter 2

Solution of Non-Linear Equations

Introduction

Bisection Method

Regula-Falsi Method

Method of iteration

Newton - Raphson Method

Secant Method

Muller's Method

Graeffe's Root Squaring

Method

Bisection Method (Bolzano)

Regula-Falsi Method

**Method of false
position**

Method of Iteration

Let $x = a$ is the desired root

$$x = \phi(x)$$

Suppose x_0 is its initial approximation. The first and successive approximations to the root can be obtained as

$$\left. \begin{aligned} x_1 &= \phi(x_0) \\ x_2 &= \phi(x_1) \\ &\vdots \\ x_{n+1} &= \phi(x_n) \end{aligned} \right\}$$

Newton - Raphson Method

An approximation to the root is given by

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

Better and successive approximations $x_2, x_3, \dots, x_n$ to the root are obtained from

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

N-R Formula

Example

Find a real root of the equation $x^3 - x - 1 = 0$ using Newton - Raphson method, correct to four decimal places.

Solution

Let $f(x) = x^3 - x - 1$,

Then $f(1) = -1$,

$$f(2) = 5$$

Therefore, the root lies
in the interval $(1, 2)$.

Further $f'(x) = 3x^2 - 1$,

$$f''(x) = 6x$$

$$f(1) = -1, \quad f''(1) = 6,$$

$$f(2) = 5, \quad f''(2) = 12$$

Since $f(2)$ and $f''(2)$ are of the same sign, we choose $x_0 = 2$ as the first approximation to the root.

**The second approximation is
computed using Newton-Raphson
method as**

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 2 - \frac{5}{11} = 1.54545$$

and

$$f(x_1) = 1.14573$$

The successive approximations are

$$x_2 = 1.54545 - \frac{1.14573}{6.16525} = 1.35961, \quad f(x_2) = 0.15369$$

$$x_3 = 1.35961 - \frac{0.15369}{4.54562} = 1.32579, \quad f(x_3) = 4.60959 \times 10^{-3}$$

$$x_4 = 1.32579 - \frac{4.60959 \times 10^{-3}}{4.27316} = 1.32471, \quad f(x_4) = -3.39345 \times 10^{-5}$$

$$x_5 = 1.32471 + \frac{3.39345 \times 10^{-5}}{4.26457} = 1.324718, \quad f(x_5) = 1.823 \times 10^{-7}$$

**Hence, the required
root is 1.3247.**

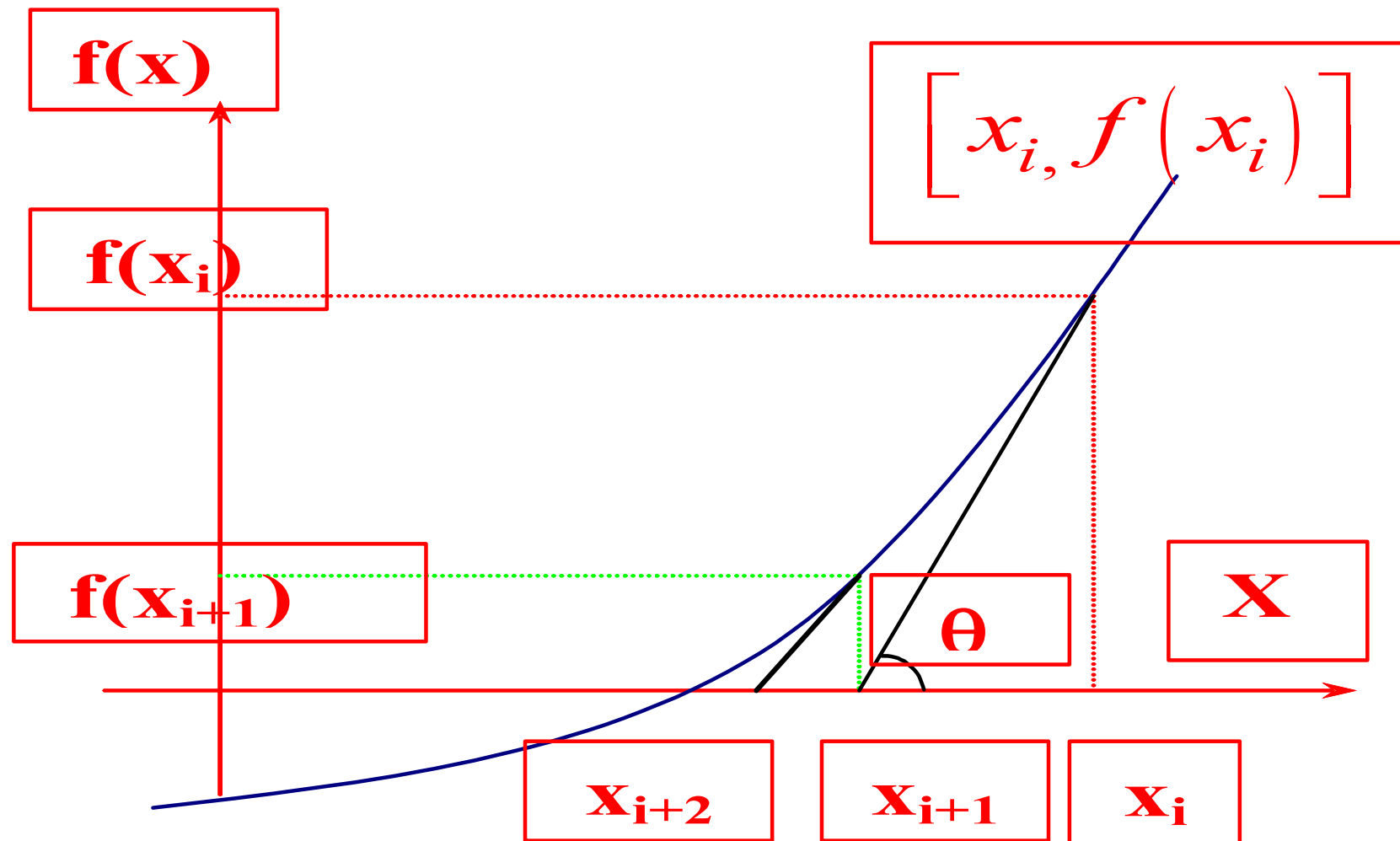
Note !

Methods such as bisection method and the false position method of finding roots of a nonlinear equation $f(x) = 0$ require bracketing of the root by two guesses. Such methods are called bracketing methods.

**These methods are
always convergent since
they are based on
reducing the interval
between the two
guesses to zero *in* on
the root.**

In the Newton-Raphson method, the root is not bracketed. Only one initial guess of the root is needed to get the iterative process started to find the root of an equation. Hence, the method falls in the category of open methods.

Newton - Raphson method is based on the principle that if the initial guess of the root of $f(x) = 0$ is at x_i , then if one draws the tangent to the curve at $f(x_i)$, the point x_{i+1} where the tangent crosses the x-axis is an improved estimate of the root



Draw backs of N-R Method

Divergence at inflection points:

If the selection of a guess or an iterated value turns out to be close to the inflection point of $f(x)$,
[where $f''(x) = 0$],
the roots start to diverge away from the root.

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

Division of zero or near zero:

If an iteration, we come across the division by zero or a near-zero number, then we get a large magnitude for the next value, x_{i+1} .

Root jumping:

In some case where the function $f(x)$ is oscillating and has a number of roots, one may choose an initial guess close to a root. However, the guesses may jump and converge to some other root.

Oscillations near local maximum and minimum:

Results obtained from N-R method may oscillate about the local max or min without converging on a root but converging on the local max or min. Eventually, it may lead to division to a number close to zero and may diverge.

Convergence of N-R Method

Let us compare the N-R formula

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

with the general iteration formula

$$x_{n+1} = \phi(x_n),$$

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

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

The iteration method converges if

$$|\phi'(x)| < 1.$$

**Therefore, N-R formula
converges, provided**

$$|f(x)f''(x)| < |f'(x)|^2$$

in the interval considered.

**Newton-Raphson formula
therefore converges,
provided the initial
approximation x_0 is chosen
sufficiently close to the root
and are continuous and
bounded in any small interval
containing the root.**

Definition

Let $x_n = \alpha + \varepsilon_n$, $x_{n+1} = \alpha + \varepsilon_{n+1}$

where α **is a root of** $f(x) = 0$.

If we can prove that $\varepsilon_{n+1} = K \varepsilon_n^p$,

where K **is a constant and** ε_n **is the error involved at the** n **- th step, while finding the root by an iterative method, then the rate of convergence of the method is** p .

The N-R method converges quadratic ally

$$x_n = \alpha + \varepsilon_n,$$

$$x_{n+1} = \alpha + \varepsilon_{n+1}$$

where α is a root of $f(x) = 0$ and ε_n is the error involved at the n -th step, while finding the root by N-R formula

$$\alpha + \varepsilon_{n+1} = \alpha + \varepsilon_n - \frac{f(\alpha + \varepsilon_n)}{f'(\alpha + \varepsilon_n)}$$

$$\varepsilon_{n+1} = \varepsilon_n - \frac{f(\alpha + \varepsilon_n)}{f'(\alpha + \varepsilon_n)} = \frac{\varepsilon_n f'(\alpha + \varepsilon_n) - f(\alpha + \varepsilon_n)}{f'(\alpha + \varepsilon_n)}$$

Using Taylor's expansion, we get

$$\varepsilon_{n+1} = \frac{1}{f'(\alpha) + \varepsilon_n f''(\alpha) + \cdots} \left\{ \varepsilon_n [f'(\alpha) + \varepsilon_n f''(\alpha) + \cdots] \right. \\ \left. - \left[f(\alpha) + \varepsilon_n f'(\alpha) + \frac{\varepsilon_n^2}{2} f''(\alpha) + \cdots \right] \right\}$$

Since α is a root, $f(\alpha) = 0$. Therefore, the above expression simplifies to

$$\begin{aligned}\varepsilon_{n+1} &= \frac{\varepsilon_n^2}{2} f''(\alpha) \frac{1}{f'(\alpha) + \varepsilon_n f''(\alpha)} \\&= \frac{\varepsilon_n^2}{2} \frac{f''(\alpha)}{f'(\alpha)} \left[1 + \varepsilon_n \frac{f''(\alpha)}{f'(\alpha)} \right]^{-1} \\&= \frac{\varepsilon_n^2}{2} \frac{f''(\alpha)}{f'(\alpha)} \left[1 - \varepsilon_n \frac{f''(\alpha)}{f'(\alpha)} \right] \\&= \frac{\varepsilon_n^2}{2} \frac{f''(\alpha)}{f'(\alpha)} + O(\varepsilon_n^3)\end{aligned}$$

On neglecting terms of order ε_n^3 and higher powers, we obtain

$$\varepsilon_{n+1} = K \varepsilon_n^2$$

Where
$$K = \frac{f''(\alpha)}{2f'(\alpha)}$$

It shows that N-R method has second order convergence or converges quadratic ally.

Example

Set up Newton's scheme of iteration for finding the square root of a positive number N .

Solution

The square root of N can be carried out as a root of the equation

$$x^2 - N = 0.$$

Let

$$f(x) = x^2 - N.$$

By Newton's method, we have

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

In this Problem

$$f(x) = x^2 - N, f'(x) = 2x.$$

Therefore

$$x_{n+1} = x_n - \frac{x_n^2 - N}{2x_n} = \frac{1}{2} \left(x_n + \frac{N}{x_n} \right)$$

Example

Evaluate $\sqrt{12}$, by Newton's formula.

Solution

Since $\sqrt{9} = 3, \sqrt{16} = 4,$

we take $x_0 = (3 + 4) / 2 = 3.5.$

$$x_1 = \frac{1}{2} \left(x_0 + \frac{N}{x_0} \right) = \frac{1}{2} \left(3.5 + \frac{12}{3.5} \right) = 3.4643$$

$$x_2 = \frac{1}{2} \left(3.4643 + \frac{12}{3.4643} \right) = 3.4641$$

$$x_3 = \frac{1}{2} \left(3.4641 + \frac{12}{3.4641} \right) = 3.4641$$

Hence $\sqrt{12} = 3.4641$.

Numerical Analysis Lecture 6

Numerical Analysis Lecture 7

Chapter 2

Solution of Non-Linear Equations

Introduction

Bisection Method

Regula-Falsi Method

Method of iteration

Newton - Raphson Method

Secant Method

Muller's Method

Graeffe's Root Squaring

Method

Secant Method

We choose x_0, x_1 close to the root 'a' of $f(x) = 0$ such that $f(x_0) \neq f(x_1)$

As a next approximation x_2 is obtained as the point of intersection of $y = 0$ and the chord passing through the points

$(x_0, f(x_0)), (x_1, f(x_1))$.

$$y - f(x_0) = \frac{f(x_1) - f(x_0)}{x_1 - x_0} (x - x_0),$$

Putting $y = 0$ we get

$$x_2 = x = \frac{x_0 f(x_1) - x_1 f(x_0)}{f(x_1) - f(x_0)}$$

Generally

$$x_{n+1} = \frac{x_{n-1} f(x_n) - x_n f(x_{n-1})}{f(x_n) - f(x_{n-1})}, n = 1, 2, 3, \dots$$

This sequence converges to the root 'b' of $f(x) = 0$ i.e. $f(b) = 0$.

Convergence of Secant Method

Here the sequence is generated by the rule

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

starting with x_0 and x_1 as

$$\{x_0, x_1, \dots\}$$

It will converge to 'a', that is $f(a) = 0$

**The Secant method
converges faster
than linear and
slower than
Newton's quadratic.**

Example

Do three iterations of secant method to find the root of

$$f(x) = x^3 - 3x + 1 = 0,$$

taking

$$x_0 = 1, x_1 = 0.5$$

$$n = 1,$$

$$f(x_0) = f(1) = -1,$$

$$f(x_1) = f(0.5) = -0.375$$

$$\begin{aligned} x_2 &= \frac{x_0 f(x_1) - x_1 f(x_0)}{f(x_1) - f(x_0)} \\ &= \frac{(1)(-0.375) - (0.5)(-1)}{-0.375 - (-1)} = 0.2 \end{aligned}$$

$$n = 2,$$

$$f(x_2) = f(0.2) = 0.408$$

$$\begin{aligned} x_3 &= \frac{x_1 f(x_2) - x_2 f(x_1)}{f(x_2) - f(x_1)} \\ &= \frac{(0.5)(0.408) - 0.2(-0.375)}{0.408 - (-0.375)} = 0.3563 \end{aligned}$$

$$n = 3,$$

$$f(x_3) = f(0.3563) = -0.02367$$

$$x_4 = \frac{x_2 f(x_3) - x_3 f(x_2)}{f(x_3) - f(x_2)}$$

$$= \frac{(0.2)f(0.3563) - 0.3563f(0.2)}{f(0.3563) - f(0.2)}$$

$$= \frac{(0.2)(-0.02367) - (0.3563)(0.408)}{-0.02367 - 0.408} = 0.3477$$

$$x_5 = 0.3473, f(x_5) = -0.0000096$$

and $|x_5 - x_4| = 0.0004$.

Though x_5 is not the true root, yet this is good approximation to the root and convergence is faster than bisection.

Comparison:

In secant method, we do not check whether the root lies in between two successive approximates X_{n-1} , and X_n .

This checking was imposed after each iteration, in Regula –Falsi method.

Muller's Method

In *Muller's method*, $f(x) = 0$ is approximated by a second degree polynomial; that is by a quadratic equation that fits through three points in the vicinity of a root. The roots of this quadratic equation are then approximated to the roots of the equation $f(x) = 0$.

This method is iterative in nature and does not require the evaluation of derivatives as in Newton-Raphson method. This method can also be used to determine both real and complex roots of $f(x) = 0$.

**Suppose, x_{i-2}, x_{i-1}, x_i
be any three distinct
approximations to a root
of $f(x) = 0$.**

$$f(x_{i-2}) = f_{i-2}, f(x_{i-1}) = f_{i-1}$$

$$f(x_i) = f_i.$$

Noting that any three distinct points in the (x, y) -plane uniquely, determine a polynomial of second degree.

A general polynomial of second degree is given by

$$f(x) = ax^2 + bx + c$$

Suppose, it passes through the points

$$(x_{i-2}, f_{i-2}), (x_{i-1}, f_{i-1}), (x_i, f_i)$$

**then the following equations will
be satisfied**

$$ax_{i-2}^2 + bx_{i-2} + c = f_{i-2}$$

$$ax_{i-1}^2 + bx_{i-1} + c = f_{i-1}$$

$$ax_i^2 + bx_i + c = f_i$$

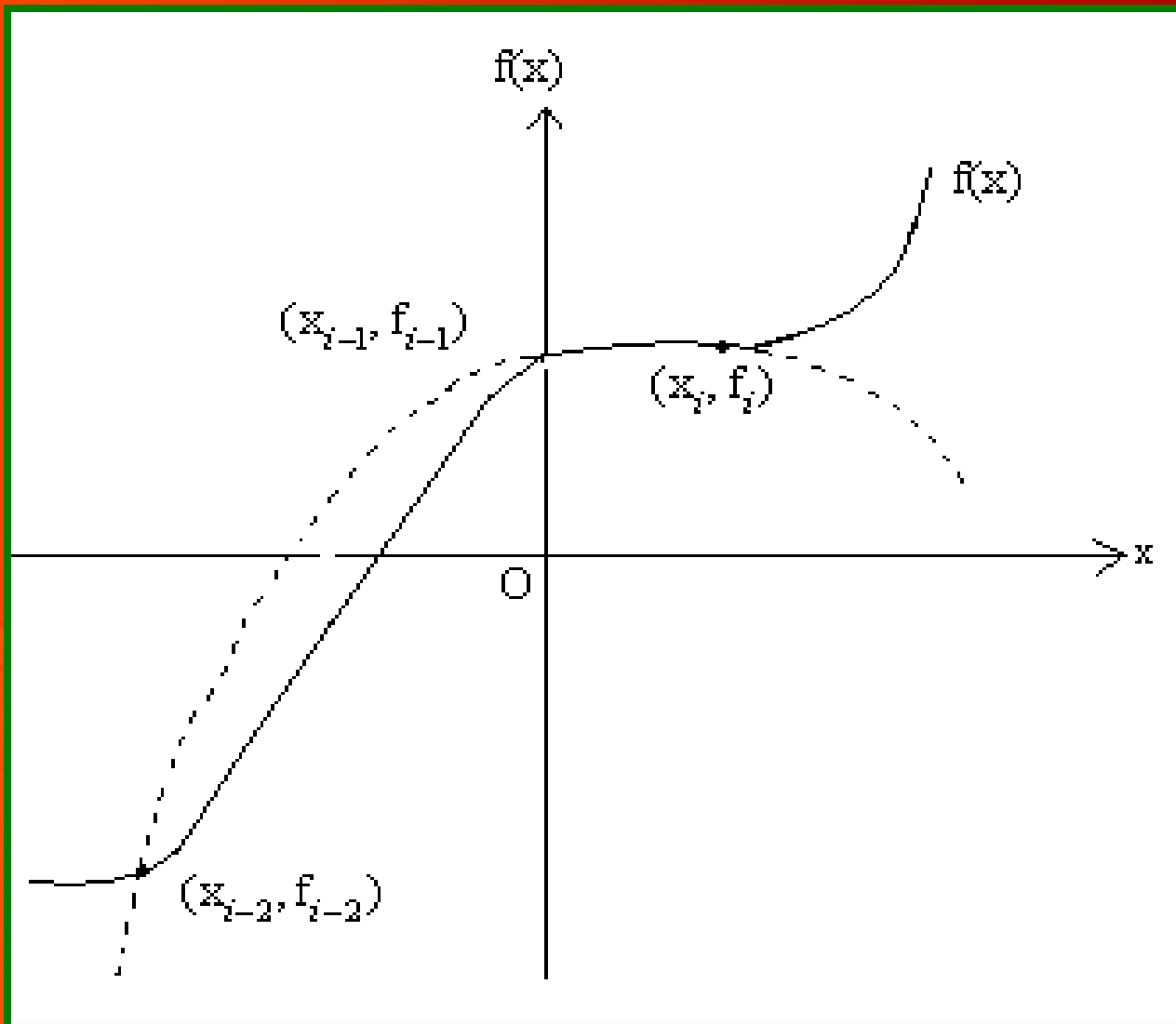


Fig: Quadratic polynomial.

Eliminating a, b, c , we obtain

$$\begin{vmatrix} x^2 & x & 1 & f \\ x_{i-2}^2 & x_{i-2} & 1 & f_{i-2} \\ x_{i-1}^2 & x_{i-1} & 1 & f_{i-1} \\ x_i^2 & x_i & 1 & f_i \end{vmatrix} = 0$$

which can be written as

$$\begin{aligned} f &= \frac{(x - x_{i-1})(x - x_i)}{(x_{i-2} - x_{i-1})} f_{i-2} \\ &+ \frac{(x - x_{i-2})(x - x_i)}{(x_{i-1} - x_{i-2})(x_{i-1} - x_i)} f_{i-1} \\ &+ \frac{(x - x_{i-2})(x - x_{i-1})}{(x_i - x_{i-2})(x_i - x_{i-1})} f_i \end{aligned}$$

That was a second degree polynomial.

Now, introducing the notation

$$h = x - x_i,$$

$$h_i = x_i - x_{i-1},$$

$$h_{i-1} = x_{i-1} - x_{i-2}$$

The above equation can be written as

The above equation can be written as

$$\begin{aligned} f = & \frac{(h + h_i)h}{-h_{i-1}(-h_{i-1} - h_i)} f_{i-2} \\ & + \frac{(h + h_i + h_{i-1})h}{(h_{i-1})(-h_i)} f_{i-1} \\ & + \frac{(h + h_i + h_{i-1})(h + h_i)}{(h_i + h_{i-1})h_i} f_i \end{aligned}$$

We further define

$$\lambda = \frac{h}{h_i} = \frac{x - x_i}{x_i - x_{i-1}}$$

$$\lambda_i = \frac{h_i}{h_{i-1}}$$

$$\delta_i = 1 + \lambda_i$$

**With these substitution we
get a simplified Equation as**

$$f = \frac{1}{\delta_i} [\lambda(\lambda + 1) \lambda_i^2 f_{i-2}$$

$$- \lambda(\lambda + 1 + \lambda_i^{-1}) \lambda_i \delta_i f_{i-1}$$

$$+ (\lambda + 1 + \lambda_i^{-1}) \lambda_i f_i]$$

Or

$$\begin{aligned} f &= \lambda^2 (f_{i-2} \lambda_i^2 - f_{i-1} \lambda_i \delta_i + f_i \lambda_i) \delta_i^{-1} \\ &\quad + \lambda [f_{i-2} \lambda_i^2 - f_{i-1} \delta_i^2 \\ &\quad + f_i (\lambda_i + \delta_i)] \delta_i^{-1} + f_i \end{aligned}$$

To compute λ , set $f=0$, we obtain

$$\lambda_i(f_{i-2}\lambda_i - f_{i-1}\delta_i + f_i)\lambda^2 + g_i\lambda + \delta_i f_i = 0$$

where $g_i = f_{i-2}\lambda_i^2 - f_{i-1}\delta_i^2 + f_i(\lambda_i + \delta_i)$

A direct solution will lead to loss of accuracy and therefore to obtain

max accuracy we rewrite as:

$$\frac{f_i\delta_i}{\lambda^2} + \frac{g_i}{\lambda} + \lambda_i(f_{i-2}\lambda_i - f_{i-1}\delta_i + f_i) = 0$$

so that,

or
$$\frac{1}{\lambda} = \frac{-g_i \pm [g_i^2 - 4f_i\delta_i\lambda_i(f_{i-2}\lambda_i - f_{i-1}\delta_i + f_i)]^{1/2}}{2f_i\delta_i}$$

$$\lambda = \frac{-2f_i\delta_i}{g_i \pm [g_i^2 - 4f_i\delta_i\lambda_i(f_{i-2}\lambda_i - f_{i-1}\delta_i + f_i)]^{1/2}}$$

Here, the positive sign must be so chosen that the denominator becomes largest in magnitude.

**we can get a better
approximation to the root, by
using**

$$x_{i+1} = x_i + h_i \lambda$$

Example

Find the root of the equation $x^3 - x - 1 = 0$ using Muller's method.

Numerical Analysis Lecture 7

Numerical Analysis Lecture 8

Chapter 2

Solution of Non-Linear Equations

Introduction

Bisection Method

Regula-Falsi Method

Method of iteration

Newton - Raphson Method

Secant Method

Muller's Method

Graeffe's Root Squaring

Method

Muller's Method

**Suppose, x_{i-2}, x_{i-1}, x_i
be any three distinct
approximations to a root
of $f(x) = 0$.**

$$f(x_{i-2}) = f_{i-2}, f(x_{i-1}) = f_{i-1}$$

$$f(x_i) = f_i.$$

Noting that any three distinct points in the (x, y) -plane uniquely, determine a polynomial of second degree.

A general polynomial of second degree is given by

$$f(x) = ax^2 + bx + c$$

Suppose, it passes through the points

$$(x_{i-2}, f_{i-2}), (x_{i-1}, f_{i-1}), (x_i, f_i)$$

**then the following equations will
be satisfied**

$$ax_{i-2}^2 + bx_{i-2} + c = f_{i-2}$$

$$ax_{i-1}^2 + bx_{i-1} + c = f_{i-1}$$

$$ax_i^2 + bx_i + c = f_i$$

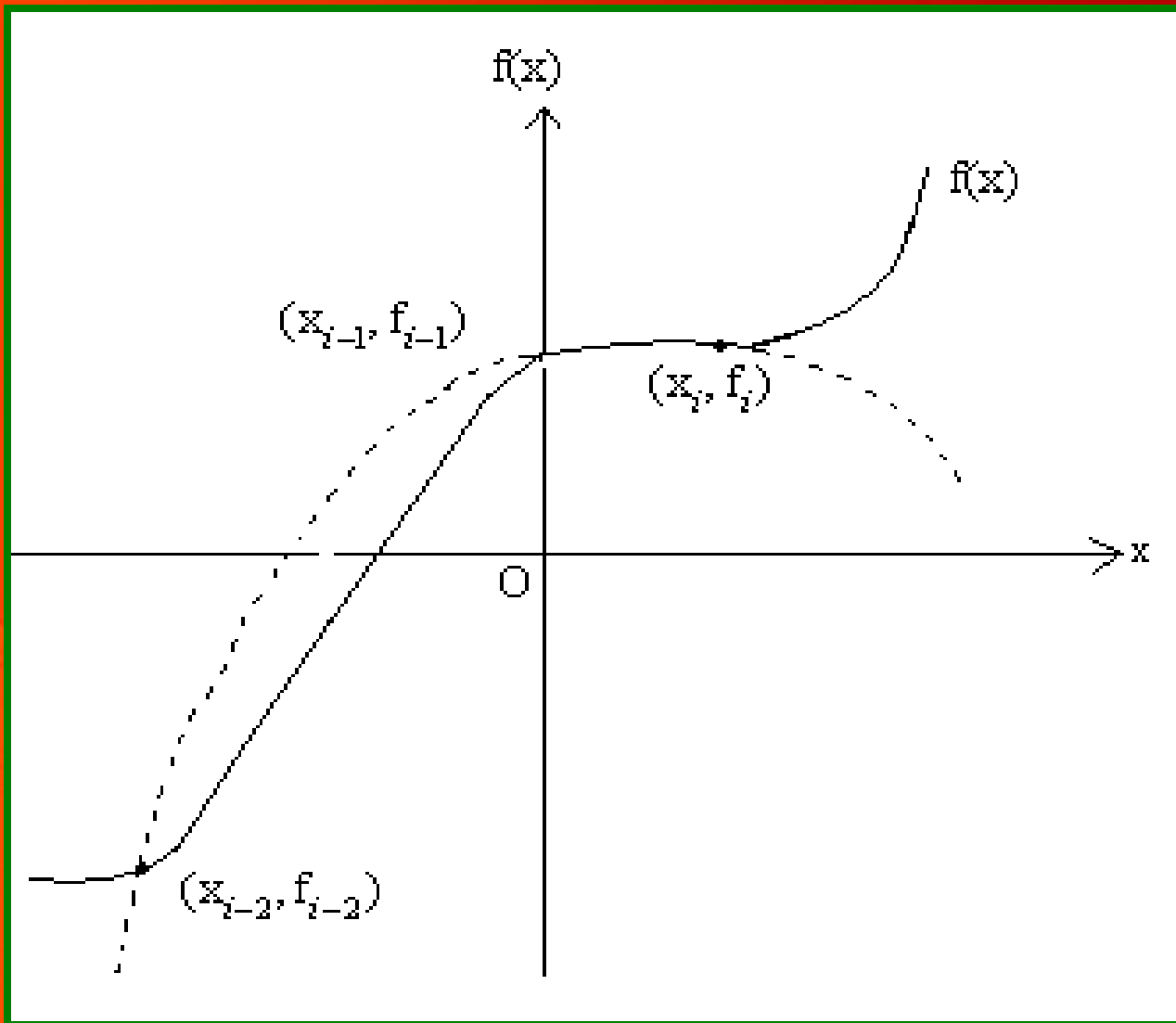


Fig: Quadratic polynomial.

Eliminating a, b, c , we obtain

$$\begin{vmatrix} x^2 & x & 1 & f \\ x_{i-2}^2 & x_{i-2} & 1 & f_{i-2} \\ x_{i-1}^2 & x_{i-1} & 1 & f_{i-1} \\ x_i^2 & x_i & 1 & f_i \end{vmatrix} = 0$$

which can be written as

$$\begin{aligned} f &= \frac{(x - x_{i-1})(x - x_i)}{(x_{i-2} - x_{i-1})} f_{i-2} \\ &+ \frac{(x - x_{i-2})(x - x_i)}{(x_{i-1} - x_{i-2})(x_{i-1} - x_i)} f_{i-1} \\ &+ \frac{(x - x_{i-2})(x - x_{i-1})}{(x_i - x_{i-2})(x_i - x_{i-1})} f_i \end{aligned}$$

That was a second degree polynomial.

Now, introducing the notation

$$h = x - x_i,$$

$$h_i = x_i - x_{i-1},$$

$$h_{i-1} = x_{i-1} - x_{i-2}$$

The above equation can be written as

The above equation can be written as

$$\begin{aligned} f = & \frac{(h + h_i)h}{-h_{i-1}(-h_{i-1} - h_i)} f_{i-2} \\ & + \frac{(h + h_i + h_{i-1})h}{(h_{i-1})(-h_i)} f_{i-1} \\ & + \frac{(h + h_i + h_{i-1})(h + h_i)}{(h_i + h_{i-1})h_i} f_i \end{aligned}$$

We further define

$$\lambda = \frac{h}{h_i} = \frac{x - x_i}{x_i - x_{i-1}}$$

$$\lambda_i = \frac{h_i}{h_{i-1}}$$

$$\delta_i = 1 + \lambda_i$$

**With these substitution we
get a simplified Equation as**

$$f = \frac{1}{\delta_i} [\lambda(\lambda + 1) \lambda_i^2 f_{i-2}$$

$$- \lambda(\lambda + 1 + \lambda_i^{-1}) \lambda_i \delta_i f_{i-1}$$

$$+ (\lambda + 1 + \lambda_i^{-1}) \lambda_i f_i]$$

Or

$$\begin{aligned} f &= \lambda^2 (f_{i-2} \lambda_i^2 - f_{i-1} \lambda_i \delta_i + f_i \lambda_i) \delta_i^{-1} \\ &\quad + \lambda [f_{i-2} \lambda_i^2 - f_{i-1} \delta_i^2 \\ &\quad + f_i (\lambda_i + \delta_i)] \delta_i^{-1} + f_i \end{aligned}$$

To compute λ , set $f=0$, we obtain

$$\lambda_i(f_{i-2}\lambda_i - f_{i-1}\delta_i + f_i)\lambda^2 + g_i\lambda + \delta_i f_i = 0$$

where $g_i = f_{i-2}\lambda_i^2 - f_{i-1}\delta_i^2 + f_i(\lambda_i + \delta_i)$

A direct solution will lead to loss of accuracy and therefore to obtain

max accuracy we rewrite as:

$$\frac{f_i\delta_i}{\lambda^2} + \frac{g_i}{\lambda} + \lambda_i(f_{i-2}\lambda_i - f_{i-1}\delta_i + f_i) = 0$$

so that,

or
$$\frac{1}{\lambda} = \frac{-g_i \pm [g_i^2 - 4f_i\delta_i\lambda_i(f_{i-2}\lambda_i - f_{i-1}\delta_i + f_i)]^{1/2}}{2f_i\delta_i}$$

$$\lambda = \frac{-2f_i\delta_i}{g_i \pm [g_i^2 - 4f_i\delta_i\lambda_i(f_{i-2}\lambda_i - f_{i-1}\delta_i + f_i)]^{1/2}}$$

Here, the positive sign must be so chosen that the denominator becomes largest in magnitude.

**we can get a better
approximation to the root, by
using**

$$x_{i+1} = x_i + h_i \lambda$$

Example

Do two iterations of Muller's method to solve

$$x^3 - 3x + 1 = 0$$

starting with

$$x_0 = 0.5, \quad x_2 = 0, \quad x_1 = 1$$

Solution

$$f(x_0) = f_0 = (0.5)^3 - 3(0.5) + 1 = -0.375$$

$$f(x_1) = f_1 = (1)^3 - 3(1) + 1 = -1$$

$$f(x_2) = f_2 = 0 - 3 \times 0 + 1 = +1$$

$$c = f_0 = -0.375$$

$$h_1 = x_1 - x_0 = 0.5$$

$$h_2 = x_0 - x_2 = 0.5$$

$$a = \frac{h_2 f_1 - (h_1 + h_2) f_0 + h_1 f_2}{h_1 h_2 (h_1 + h_2)}$$

$$= \frac{(0.5)(-1) - (-0.375) + (0.5)}{0.25} = 1.5$$

$$b = \frac{f_1 - f_0 - ah_1^2}{h_1} = -2$$

$$\therefore x = x_0 - \frac{2c}{+b - \sqrt{b^2 - 4ac}}$$

$$= 0.5 - \frac{2(-0.375)}{-2 - \sqrt{4 + 4(1.5)(0.375)}}$$

$$= 0.5 - \frac{-0.75}{-2 - \sqrt{4 + 2.25}} = 0.333333 \prec 0.5$$

Take

$$x_2 = 0, x_0 = 0.333333, x_1 = 0.5$$

$$h_1 = x_1 - x_0 = 0.166667, h_2 = x_0 - x_2 = 0.333333$$

$$c = f_0 = f(0.333333)$$

$$= x_0^3 - 3x_0 + 1 = 0.037046$$

$$f_1 = x_1^3 - 3x_1 + 1 = -0.375$$

$$f_2 = x_2^3 - 3x_2 + 1 = 1$$

$$a = \frac{h_2 f_1 - (h_1 + h_2) f_0 + h_1 f_2}{h_1 h_2 (h_1 + h_2)} = \frac{0.023148}{0.027778}$$

$$= 0.8333$$

$$b = \frac{f_1 - f_0 - ah_1^2}{h_1} = -2.6$$

$$\begin{aligned}x &= x_0 - \frac{2c}{b - \sqrt{b^2 - 4ac}} \\&= 0.3333333 - \frac{0.074092}{-5.2236} \\&= 0.3475 \succ 0.333333 = x_0\end{aligned}$$

For third iteration take,

$$x_2 = 0.333333, \quad x_0 = 0.3475, \quad x_1 = 0.5$$

Graeffe's Root Squaring Method

GRAEFFE'S ROOT SQUARING METHOD is particularly attractive for finding all the roots of a polynomial equation.

Consider a polynomial of third degree

$$f(x) = a_0 + a_1x + a_2x^2 + a_3x^3$$

$$f(x) = a_0 + a_1x + a_2x^2 + a_3x^3$$

$$f(-x) = a_0 - a_1x + a_2x^2 - a_3x^3$$

$$f(x)f(-x) = a_3^2x^6 - (a_2^2 - 2a_1a_3)x^4 \\ + (a_1^2 - 2a_0a_2)x^2 - a_0^2$$

$$f(x)f(-x) = a_3^2t^3 - (a_2^2 - 2a_1a_3)t^2 \\ + (a_1^2 - 2a_0a_2)t - a_0^2$$

The roots of this equation are squares or 2^i ($i = 1$), powers of the original roots. Here $i = 1$ indicates that squaring is done once.

The same equation can again be squared and this squaring process is repeated as many times as required. After each squaring, the coefficients become large and overflow is possible as i increases.

Suppose, we have squared the given polynomial ' i ' times, then we can estimate the value of the roots by evaluating 2^i root of

$$\left| \frac{a_i}{a_{i-1}} \right|, \quad i = 1, 2, \dots, n$$

where n is the degree of the given polynomial.

The proper sign of each root can be determined by recalling the original equation. This method fails, when the roots of the given polynomial are repeated.

Example

Using Graeffe root squaring method, find all the roots of the equation

$$x^3 - 6x^2 + 11x - 6 = 0$$

Solution

Using Graeffe root squaring method, the first three squared polynomials are as under:

For $i = 1$, the polynomial is

$$\begin{aligned} & x^3 - (36 - 22)x^2 + (121 - 72)x - 36 \\ &= x^3 - 14x^2 + 49x - 36 \end{aligned}$$

For $i = 2$, the polynomial is

$$\begin{aligned} & x^3 - (196 - 98)x^2 + (2401 - 1008)x - 1296 \\ &= x^3 - 98x^2 + 1393x - 1296 \end{aligned}$$

For $i = 3$, the polynomial is

$$\begin{aligned} & x^3 - (9604 - 2786)x^2 + (1940449 - 254016)x - 1679616 \\ &= x^3 - 6818x^2 + 16864333x - 1679616 \end{aligned}$$

The roots of polynomial are

$$\sqrt{\frac{36}{49}} = 0.85714, \quad \sqrt{\frac{49}{14}} = 1.8708, \quad \sqrt{\frac{14}{1}} = 3.7417$$

Similarly, the roots of polynomial (2) are

$$\sqrt[4]{\frac{1296}{1393}} = 0.9821, \quad \sqrt[4]{\frac{1393}{98}} = 1.9417, \quad \sqrt[4]{\frac{98}{1}} = 3.1464$$

Still better estimates of the roots obtained from polynomial (3) are

$$\sqrt[8]{\frac{1679616}{1686433}} = 0.99949, \quad \sqrt[8]{\frac{1686433}{6818}} = 1.99143, \quad \sqrt[8]{\frac{6818}{1}} = 3.0144$$

The exact values of the roots of the given polynomial are 1, 2 and 3.

Numerical Analysis Lecture 8

Revision Example

Obtain the Newton-Raphson extended formula

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} - \frac{1}{2} \frac{[f(x_0)]^2}{[f'(x_0)]^3} f''(x_0)$$

for finding the root of the equation $f(x) = 0$.

Solution

Expanding $f(x)$ by Taylor's series, in the neighborhood of x_0 , we obtain after retaining the first order term only

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

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

This is the first approximation to the root.

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

Expanding $f(x)$ by Taylor's series and retaining up to second order term,

$$0 = f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{(x - x_0)^2}{2} f''(x_0)$$

Therefore, $f(x_1) = f(x_0) + (x_1 - x_0)f'(x_0) + \frac{(x_1 - x_0)^2}{2} f''(x_0) = 0$

This can also be written as

$$f(x_0) + (x_1 - x_0)f'(x_0) + \frac{1}{2} \frac{[f(x_0)]^2}{[f'(x_0)]^2} f''(x_0) = 0$$

Thus, the Newton-Raphson extended formula is given by

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} - \frac{1}{2} \frac{[f(x_0)]^2}{[f'(x_0)]^3} f''(x_0)$$

This is also known as Chebyshev's formula of third order

Numerical Analysis Lecture 8

Numerical Analysis Lecture 9

Chapter 3

Solution of Linear System of Equations

and Matrix Inversion

Introduction

Gaussian Elimination

Gauss-Jordan

Elimination

Crout's Reduction

Jacobi's

Gauss- Seidal Iteration

Relaxation

Matrix Inversion

The solution of the system of equations gives n unknown values $x_1, x_2, \dots, x_n$, which satisfy the system simultaneously.

If $m > n$, the system usually will have an infinite number of solutions.

If $|A| \neq 0$

**then the system will have a
unique solution.**

If $|A| = 0$,

**then there exists no
solution.**

Numerical methods for finding the solution of the system of equations are classified as direct and iterative methods. In direct methods, we get the solution of the system after performing all the steps involved in the procedure. The direct methods consist of elimination methods and decomposition methods.

Under elimination methods, we consider, Gaussian elimination and Gauss-Jordan elimination methods

Crout's reduction also known as Cholesky's reduction is considered under decomposition methods.

Under iterative methods, the initial approximate solution is assumed to be known and is improved towards the exact solution in an iterative way. We consider Jacobi, Gauss-Seidel and relaxation methods under iterative methods.

Gaussian Elimination Method

In this method, the solution to the system of equations is obtained in two stages.

i) the given system of equations is reduced to an equivalent upper triangular form using elementary transformations

ii) the upper triangular system is solved using back substitution procedure

This method is explained by considering a system of n equations in n unknowns in the form as follows

$$\left. \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{array} \right\}$$

Stage I: Substituting the value of x_1 from first equation into the rest

$$\left. \begin{array}{l} x_1 + a'_{12}x_2 + \cdots + a'_{1n}x_n = b'_1 \\ a'_{22}x_2 + \cdots + a'_{2n}x_n = b'_2 \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ a'_{n2}x_2 + \cdots + a'_{nn}x_n = b'_n \end{array} \right\}$$

Now, the last $(n - 1)$ equations are independent of x_1 , that is, x_1 is eliminated from the last $(n - 1)$ equations.

This procedure is repeated and x_2 is eliminated from 3rd, 4th, ..., n -th equations

The same procedure is repeated till the given system assumes the upper triangular form:

$$\left. \begin{aligned}
 c_{11}x_1 + c_{12}x_2 + \cdots + c_{1n}x_n &= d_1 \\
 c_{22}x_2 + \cdots + c_{2n}x_n &= d_2 \\
 \vdots &\vdots \\
 c_{nn}x_n &= d_n
 \end{aligned} \right\}$$

Stage II: The values of the unknowns are determined by backward substitution.

First x_n is found from the last equation and then substitution this value of x_n in the preceding equation will give the value of x_{n-1} . Continuing this way, we can find the values of all other unknowns

Example

Solve the following system of equations using Gaussian elimination method

$$2x + 3y - z = 5$$

$$4x + 4y - 3z = 3$$

$$-2x + 3y - z = 1$$

Solution

Stage I (*Reduction to upper-triangular form*):

Divide first equation by 2 and then subtract the resulting equation (multiplied by 4 and -2) from the 2nd and 3rd equations respectively. Thus, we eliminate x from the 2nd and 3rd equations.

The resulting new system is given by

$$\left. \begin{array}{l} x + \frac{3}{2}y - \frac{z}{2} = \frac{5}{2} \\ -2y - z = -7 \\ 6y - 2z = 6 \end{array} \right\}$$

Now, we divide the 2nd equation by -2 and eliminate y from the last equation and the modified system is given by

$$\left. \begin{array}{rcl} x + \frac{3}{2}y - \frac{z}{2} & = & \frac{5}{2} \\ y + \frac{z}{2} & = & \frac{7}{2} \\ -5z & = & -15 \end{array} \right\}$$

Stage II (Backward substitution):

From the last equation, we get

$$z = 3$$

using this value of z, the second equation gives

$$y = \frac{7}{2} - \frac{3}{2} = 2$$

Putting these values of y and z in the first equation, we get

$$x = \frac{5}{2} + \frac{3}{2} - 3 = 1$$

Thus, the solution of the given system is given by

$$x = 1, y = 2, z = 3$$

Partial and Full Pivoting

The Gaussian elimination method fails if any one of the pivot elements becomes zero. In such a situation, we rewrite the equations in a different order to avoid zero pivots. Changing the order of equations is called pivoting.

Partial pivoting

if the pivot happens to be zero, then the *i-th* column elements are searched for the numerically largest element. Let the *j-th* row ($j > i$) contains this element, then we interchange the *i-th* equation with the *j-th* equation and proceed for elimination. This process is continued whenever pivots become zero during elimination.

For example, let us examine the solution of the following simple system

$$10^{-5} x_1 + x_2 = 1$$

$$x_1 + x_2 = 2$$

Using Gaussian elimination method with and without partial pivoting, assuming that we require the solution accurate to only four decimal places. The solution by Gaussian elimination gives $x_1 = 0$, $x_2 = 1$.

If we use partial pivoting, the system takes the form

$$x_1 + x_2 = 2$$

$$10^{-5} x_1 + x_2 = 1$$

Using Gaussian elimination method, the solution is found to be $x_1 = 1$, $x_2 = 1$, which is a meaningful and perfect result.

In full pivoting (or complete pivoting), we interchange rows and columns, such that the largest element in the matrix of the variables also get changed. Full pivoting, in fact, is more complicated than the partial pivoting. Partial pivoting is preferred for hand computation.

The general form of a system of m linear equations in n unknowns $x_1, x_2, x_3, \dots, x_n$ can be represented in matrix form as under:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Using matrix notation, the above system can be written in compact form as

$$[A](X) = (B)$$

Example

Solve the system of equations by Gaussian elimination method with partial pivoting.

$$x + y + z = 7$$

$$3x + 3y + 4z = 24$$

$$2x + y + 3z = 16$$

Solution

In matrix notation, the given system can be written as

$$\begin{bmatrix} 1 & 1 & 1 \\ 3 & 3 & 4 \\ 2 & 1 & 3 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 7 \\ 24 \\ 16 \end{pmatrix}$$

To start with, we observe that the pivot element

$$a_{11} = 1 (\neq 0).$$

However, a glance at the first column reveals that the numerically largest element is 3 which is in second row.

Hence R_{12}

Thus the given equation takes the form after partial pivoting

$$\begin{bmatrix} 3 & 3 & 4 \\ 1 & 1 & 1 \\ 2 & 1 & 3 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 24 \\ 7 \\ 16 \end{pmatrix}$$

Stage I (Reduction to upper triangular form):

$$\begin{bmatrix} 1 & 1 & \frac{4}{3} \\ & -1 & \frac{1}{3} \\ & & -\frac{1}{3} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 8 \\ 0 \\ -1 \end{bmatrix}$$

Stage II (Back substitution):

$$z = 3$$

$$-y + 1 = 0 \quad \text{or} \quad y = 1$$

$$x + 1 + 4 = 8 \quad \text{or} \quad x = 3$$

Example

Solve the following system of equations by Gaussian elimination method with partial pivoting

$$0x_1 + 4x_2 + 2x_3 + 8x_4 = 24$$

$$4x_1 + 10x_2 + 5x_3 + 4x_4 = 32$$

$$4x_1 + 5x_2 + 6.5x_3 + 2x_4 = 26$$

$$9x_1 + 4x_2 + 4x_3 + 0x_4 = 21$$

Solution

In matrix notation, the given system can be written as

$$\begin{bmatrix} 0 & 4 & 2 & 8 \\ 4 & 10 & 5 & 4 \\ 4 & 5 & 6.5 & 2 \\ 9 & 4 & 4 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 24 \\ 32 \\ 26 \\ 21 \end{bmatrix}$$

To start with, we observe that the pivot row, that is, the first row has a zero pivot element ($a_{11} = 0$). This row should be interchanged with any row following it, which on becoming a pivot row should not have a zero pivot element.

While interchanging rows it is better to interchange the first and fourth rows, which is called partial pivoting and get,

$$\begin{bmatrix} 9 & 4 & 4 & 0 \\ 4 & 10 & 5 & 4 \\ 4 & 5 & 6.5 & 2 \\ 0 & 4 & 2 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 21 \\ 32 \\ 26 \\ 24 \end{bmatrix}$$

***Stage 1* (Reduction to upper-triangular form):**

$$\begin{bmatrix} 1 & \frac{4}{9} & \frac{4}{9} & 0 \\ 0 & 8.2222 & 3.2222 & 4 \\ 0 & 3.2222 & 4.7222 & 2 \\ 0 & 4 & 2 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2.3333 \\ 22.6666 \\ 16.6666 \\ 24 \end{bmatrix}$$

$$\begin{bmatrix} 1 & \frac{4}{9} & \frac{4}{9} & 0 \\ 0 & 1 & 0.3919 & 0.4865 \\ 0 & 0 & 3.4594 & 0.4324 \\ 0 & 0 & 0.4324 & 6.0540 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2.333 \\ 2.7568 \\ 7.78836 \\ 12.9728 \end{bmatrix}$$

$$\begin{bmatrix} 1 & \frac{4}{9} & \frac{4}{9} & 0 \\ 0 & 1 & 0.3919 & 0.4865 \\ 0 & 0 & 1 & 0.1250 \\ 0 & 0 & 0 & 5.999 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2.3333 \\ 2.7568 \\ 2.2500 \\ 11.9999 \end{bmatrix}$$

Stage II

Back substitution

$$x_4 = 2.000$$

$$x_3 = 2.0000$$

$$x_2 = 1.0000,$$

$$x_1 = 1.0000$$

$$x_1 = 1.0, \quad x_2 = 1.0, \quad x_3 = 2.0, \quad x_4 = 2.0$$

Numerical Analysis Lecture 9

Numerical

Analysis

Lecture 10

Chapter 3

Solution of Linear System of Equations and Matrix Inversion

Introduction

Gaussian Elimination

Gauss-Jordan Elimination

Crout's Reduction

Jacobi's Gauss-Seidal Iteration

Relaxation

Matrix Inversion

Gauss–Jordan Elimination Method

This method is a variation of Gaussian elimination method. In this method, the elements above and below the diagonal are simultaneously made zero

That is a given system is reduced to an equivalent diagonal form using elementary transformations. Then the solution of the resulting diagonal system is obtained.

Sometimes, we normalize the pivot row with respect to the pivot element, before elimination. Partial pivoting is also used whenever the pivot element becomes zero.

Example

**Solve the system of equations
using Gauss-Jordan
elimination method:**

$$\left. \begin{array}{l} x + 2y + z = 8 \\ 2x + 3y + 4z = 20 \\ 4x + 3y + 2z = 16 \end{array} \right\}$$

In matrix notation, the given system can be written as

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 4 \\ 4 & 3 & 2 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 8 \\ 20 \\ 16 \end{pmatrix}$$

(-2) R1 +R2 and (-4) R1+R3

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & -1 & 2 \\ 0 & -5 & -2 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 8 \\ 4 \\ -16 \end{pmatrix}$$

Now, we eliminate y from the first and third rows using the second row. Thus, we get

$$\begin{bmatrix} 1 & 0 & 5 \\ 0 & -1 & 2 \\ 0 & 0 & -12 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 16 \\ 4 \\ -36 \end{pmatrix}$$

**Before, eliminating z
from the first and
second row, normalize
the third row with
respect to the pivot
element, we get**

$$\begin{bmatrix} 1 & 0 & 5 \\ 0 & -1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 16 \\ 4 \\ 3 \end{pmatrix}$$

Using the third row of above Equation , and eliminating z from the first and second rows, we obtain

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix}$$

The solution is

$$x = 1,$$

$$y = 2,$$

$$z = 3.$$

Crout's Reduction Method

Here the coefficient matrix $[A]$ of the system of equations is decomposed into the product of two matrices $[L]$ and $[U]$, where $[L]$ is a lower-triangular matrix and $[U]$ is an upper-triangular matrix with 1's on its main diagonal.

**For the purpose of illustration,
consider a general matrix in
the form**

$$[L][U] = [A]$$

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

The sequence of steps for getting [L] and [U] are given below:

Step I: Multiplying all the rows of [L] by the first column of [U], we get

$$l_{11} = a_{11}, \quad l_{21} = a_{21}, \quad l_{31} = a_{31}$$

Thus, we observe that the first column of [L] is same as the first column of [A].

Step II: Multiplying the first row of [L] by the second and third columns of [U], we obtain

or
$$l_{11}u_{12} = a_{12}, \quad l_{11}u_{13} = a_{13}$$

$$u_{12} = \frac{a_{12}}{l_{11}}, \quad u_{13} = \frac{a_{13}}{l_{11}}$$

Thus, the first row of [U] is obtained. We continue this process, getting alternately the column of [L] and a row of [U].

Step III: Multiply the second and third rows of $[L]$ by the second column of $[U]$ to get

$$l_{21}u_{12} + l_{22} = a_{22}, \quad l_{31}u_{12} + l_{32} = a_{32}$$

which gives

$$l_{22} = a_{22} - l_{21}u_{12}, \quad l_{32} = a_{32} - l_{31}u_{12}$$

Step IV: Now, multiply the second row of [L] by the third column of [U] which yields

$$l_{21}u_{13} + l_{22}u_{23} = a_{23}$$

$$u_{23} = \frac{a_{23} - l_{21}u_{13}}{l_{22}}$$

Step V: Lastly, we multiply the third row of [L] by the third column of [U] and get

$$l_{31}u_{13} + l_{32}u_{23} + l_{33} = a_{33}$$

Which gives

$$l_{33} = a_{33} - l_{31}u_{13} - l_{32}u_{23}$$

**Thus, the above five steps
determine [L] and [U].
This algorithm can be
generalized to any linear
system of order n .**

Consider a system of equations

$$\left. \begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned} \right\}$$

In matrix notation as $[A](X) = (B)$.

Let $[A] = [L] [U]$, then we get,

$$[L][U](X) = (B)$$

**Substituting $[U] (X) = (Z)$ in Eq.
we obtain $[L] (Z) = (B)$**

$$\left. \begin{aligned} l_{11}z_1 &= b_1 \\ l_{21}z_1 + l_{22}z_2 &= b_2 \\ l_{31}z_1 + l_{32}z_2 + l_{33}z_3 &= b_3 \end{aligned} \right\}$$

**Having computed z_1 , z_2 and z_3 ,
we can compute x_1 , x_2 , and x_3
from equation $[U] (X) = (Z)$ or
from**

$$\begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}$$

This method is also known as Cholesky reduction method. This technique is widely used in the numerical solutions of partial differential equations.

This method is very popular from computer programming point of view, since the storages space reserved for matrix $[A]$ can be used to store the elements of $[L]$ and $[U]$ at the end of computation.

**This method fails
if any**

$$a_{ii} = 0$$

Example

Solve the following system of equations by Crout's reduction method

$$5x_1 - 2x_2 + x_3 = 4$$

$$7x_1 + x_2 - 5x_3 = 8$$

$$3x_1 + 7x_2 + 4x_3 = 10$$

Solution

Let the coefficient matrix $[A]$ be written as $[L] [U]$. Thus,

$$\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 5 & -2 & 1 \\ 7 & 1 & -5 \\ 3 & 7 & 4 \end{bmatrix}$$

Step I: Multiply all the rows of [L] by the first column of [U], we get

$$l_{11} = 5, \quad l_{21} = 7, \quad l_{31} = 3$$

Step II: Multiply the first row of $[L]$ by the second and third columns of $[U]$, we have

$$l_{11}u_{12} = -2, \quad l_{11}u_{13} = 1$$

$$u_{12} = -\frac{2}{5}, \quad u_{13} = \frac{1}{5}$$

STEP III: Multiply the 2nd and 3rd rows of $[L]$ by the 2nd column of $[U]$, we get

$$\left. \begin{array}{ll} l_{21}u_{12} + l_{22} = 1 & \text{or} \quad l_{22} = 1 + \frac{14}{5} = \frac{19}{5} \\ l_{31}u_{12} + l_{32} = 7 & \text{or} \quad l_{32} = 7 + \frac{6}{5} = \frac{41}{5} \end{array} \right\}$$

STEP IV: Multiply the 2nd row of [L] by the 3rd column of [U]

$$l_{21}u_{13} + l_{22}u_{23} = -5$$

$$\frac{19}{5}u_{23} = -5 - \frac{7}{5}$$

$$u_{23} = -\frac{32}{19}$$

STEP V: Finally, multiply the 3rd row of [L] with the 3rd column of [U], we obtain

$$l_{31}u_{13} + l_{32}u_{23} + l_{33} = 4$$

$$l_{33} = \frac{327}{19}$$

Thus, the given system of equations takes the form
 $[L][U][X] = (B)$.

$$\begin{bmatrix} 5 & 0 & 0 \\ 7 & \frac{19}{5} & 0 \\ 3 & \frac{41}{5} & \frac{327}{19} \end{bmatrix} \begin{bmatrix} 1 & -\frac{2}{5} & \frac{1}{5} \\ 0 & 1 & -\frac{32}{19} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 8 \\ 10 \end{bmatrix}$$

That is,

$$\begin{bmatrix} 5 & 0 & 0 \\ 7 & \frac{19}{5} & 0 \\ 3 & \frac{41}{5} & \frac{327}{19} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 8 \\ 10 \end{bmatrix}$$

Let $[U](X) = (Z)$, then

$$[L](Z) = (4 \ 8 \ 10)^T$$

or

$$\begin{bmatrix} 5 & 0 & 0 \\ 7 & \frac{19}{5} & 0 \\ 3 & \frac{41}{5} & \frac{327}{19} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 8 \\ 10 \end{bmatrix}$$

Which gives utilizing these values of z , the Eq. becomes

$$\begin{bmatrix} 1 & -\frac{2}{5} & \frac{1}{5} \\ 0 & 1 & -\frac{32}{19} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \frac{4}{5} \\ \frac{12}{19} \\ \frac{46}{327} \end{bmatrix}$$

**By back substitution method,
we obtain**

$$x_3 = \frac{46}{327}, \quad x_2 = \frac{284}{327}, \quad x_1 = \frac{366}{327}$$

This is the required solution.

Numerical

Analysis

Lecture 10

Numerical

Analysis

Lecture 11

Chapter 3

Solution of Linear System of Equations and Matrix Inversion

Introduction

Gaussian Elimination

Gauss-Jordan Elimination

Crout's Reduction

Jacobi's

Gauss- Seidal Iteration

Relaxation

Matrix Inversion

Jacobi's Method

This is an iterative method, where initial approximate solution to a given system of equations is assumed and is improved towards the exact solution in an iterative way.

In general, when the coefficient matrix of the system of equations is a sparse matrix (many elements are zero), iterative methods have definite advantage over direct methods in respect of economy of computer memory

**Such sparse matrices
arise in computing
the numerical
solution of partial
differential equations.**

Let us consider

$$\left. \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{array} \right\}$$

In this method, we assume that the coefficient matrix $[A]$ is strictly diagonally dominant, that is, in each row of $[A]$ the modulus of the diagonal element exceeds the sum of the off-diagonal elements.

We also assume that the diagonal elements do not vanish. If any diagonal element vanishes, the equations can always be rearranged to satisfy this condition.

Now the above system of equations can be written as

$$\left. \begin{aligned} x_1 &= \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}} x_2 - \dots - \frac{a_{1n}}{a_{11}} x_n \\ x_2 &= \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}} x_1 - \dots - \frac{a_{2n}}{a_{22}} x_n \\ \vdots & \quad \quad \quad \vdots \quad \quad \quad \vdots \\ x_n &= \frac{b_n}{a_{nn}} - \frac{a_{n1}}{a_{nn}} x_1 - \dots - \frac{a_{n(n-1)}}{a_{nn}} x_{n-1} \end{aligned} \right\}$$

We shall take this solution vector $(x_1, x_2, \dots, x_n)^T$ as a first approximation to the exact solution of system. For convenience, let us denote the first approximation vector by $(x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)})$ got after taking as an initial starting vector.

Substituting this first approximation in the right-hand side of system, we obtain the second approximation to the given system in the form

$$\left. \begin{aligned}
 x_1^{(2)} &= \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}} x_2^{(1)} - \dots - \frac{a_{1n}}{a_{11}} x_n^{(1)} \\
 x_2^{(2)} &= \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}} x_1^{(1)} - \dots - \frac{a_{2n}}{a_{22}} x_n^{(1)} \\
 \vdots &\quad \quad \quad \vdots \\
 x_n^{(2)} &= \frac{b_n}{a_{nn}} - \frac{a_{n1}}{a_{nn}} x_1^{(1)} - \dots - \frac{a_{n(n-1)}}{a_{nn}} x_{n-1}^{(1)}
 \end{aligned} \right\}$$

This second approximation is substituted into the right-hand side of Equations and obtain the third approximation and so on.

**This process is
repeated and $(r+1)th$
approximation is
calculated**

$$\left. \begin{aligned}
 x_1^{(r+1)} &= \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}} x_2^{(r)} - \dots - \frac{a_{1n}}{a_{11}} x_n^{(r)} \\
 x_2^{(r+1)} &= \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}} x_1^{(r)} - \dots - \frac{a_{2n}}{a_{22}} x_n^{(r)} \\
 &\vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \\
 x_n^{(r+1)} &= \frac{b_n}{a_{nn}} - \frac{a_{n1}}{a_{nn}} x_1^{(r)} - \dots - \frac{a_{n(n-1)}}{a_{nn}} x_{n-1}^{(r)}
 \end{aligned} \right\}$$

Briefly, we can rewrite these Equations as

$$x_i^{(r+1)} = \frac{b_i}{a_{ii}} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{a_{ij}}{a_{ii}} x_j^{(r)},$$

$$r = 1, 2, \dots, \quad i = 1, 2, \dots, n$$

It is also known as method of simultaneous displacements, since no element of $x_i^{(r+1)}$ is used in this iteration until every element is computed.

A sufficient condition for convergence of the iterative solution to the exact solution

is

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|, \quad i = 1, 2, \dots, n$$

When this condition (diagonal dominance) is true, Jacobi's method converges

Example

Find the solution to the following system of equations using Jacobi's iterative method for the first five iterations:

$$83x + 11y - 4z = 95$$

$$7x + 52y + 13z = 104$$

$$3x + 8y + 29z = 71$$

Solution

$$\left. \begin{aligned} x &= \frac{95}{83} - \frac{11}{83}y + \frac{4}{83}z \\ y &= \frac{104}{52} - \frac{7}{52}x - \frac{13}{52}z \\ z &= \frac{71}{29} - \frac{3}{29}x - \frac{8}{29}y \end{aligned} \right\}$$

Taking the initial starting of solution vector as $(0,0,0)^T$, from Eq. ,we have the first approximation as

$$\begin{pmatrix} x^{(1)} \\ y^{(1)} \\ z^{(1)} \end{pmatrix} = \begin{pmatrix} 1.1446 \\ 2.0000 \\ 2.4483 \end{pmatrix}$$

**Now, using Eq. ,the
second approximation is
computed from the
equations**

$$\left. \begin{aligned} x^{(2)} &= 1.1446 - 0.1325y^{(1)} + 0.0482z^{(1)} \\ y^{(2)} &= 2.0 - 0.1346x^{(1)} - 0.25z^{(1)} \\ z^{(2)} &= 2.4483 - 0.1035x^{(1)} - 0.2759y^{(1)} \end{aligned} \right\}$$

Making use of the last two equations we get the second approximation as

$$\begin{pmatrix} x^{(2)} \\ y^{(2)} \\ z^{(2)} \end{pmatrix} = \begin{pmatrix} 0.9976 \\ 1.2339 \\ 1.7424 \end{pmatrix}$$

Similar procedure yields the third, fourth and fifth approximations to the required solution and they are tabulated as below;

Variables

Iteration number r	x	y	z
1	1.1446	2.0000	2.4483
2	0.9976	1.2339	1.7424
3	1.0651	1.4301	2.0046
4	1.0517	1.3555	1.9435
5	1.0587	1.3726	1.9655

Numerical

Analysis

Lecture 11

Numerical

Analysis

Lecture 12

Chapter 3

Solution of Linear System of Equations and Matrix Inversion

Introduction

Gaussian Elimination

Gauss-Jordan Elimination

Crout's Reduction

Jacobi's

Gauss- Seidal Iteration

Relaxation

Matrix Inversion

Jacobi's Method

This is an iterative method, where initial approximate solution to a given system of equations is assumed and is improved towards the exact solution in an iterative way.

Let us consider

$$\left. \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{array} \right\}$$

In this method, we assume that the coefficient matrix $[A]$ is strictly diagonally dominant, that is, in each row of $[A]$ the modulus of the diagonal element exceeds the sum of the off-diagonal elements.

We also assume that the diagonal elements do not vanish. If any diagonal element vanishes, the equations can always be rearranged to satisfy this condition.

Now the above system of equations can be written as

$$\left. \begin{aligned} x_1 &= \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}} x_2 - \dots - \frac{a_{1n}}{a_{11}} x_n \\ x_2 &= \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}} x_1 - \dots - \frac{a_{2n}}{a_{22}} x_n \\ \vdots & \quad \quad \quad \vdots \quad \quad \quad \vdots \\ x_n &= \frac{b_n}{a_{nn}} - \frac{a_{n1}}{a_{nn}} x_1 - \dots - \frac{a_{n(n-1)}}{a_{nn}} x_{n-1} \end{aligned} \right\}$$

$$\left. \begin{aligned}
 x_1^{(2)} &= \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}} x_2^{(1)} - \dots - \frac{a_{1n}}{a_{11}} x_n^{(1)} \\
 x_2^{(2)} &= \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}} x_1^{(1)} - \dots - \frac{a_{2n}}{a_{22}} x_n^{(1)} \\
 \vdots &\quad \quad \quad \vdots \\
 x_n^{(2)} &= \frac{b_n}{a_{nn}} - \frac{a_{n1}}{a_{nn}} x_1^{(1)} - \dots - \frac{a_{n(n-1)}}{a_{nn}} x_{n-1}^{(1)}
 \end{aligned} \right\}$$

This second approximation is substituted into the right-hand side of Equations and obtain the third approximation and so on.

**This process is
repeated and $(r+1)th$
approximation is
calculated**

$$\left. \begin{aligned}
 x_1^{(r+1)} &= \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}} x_2^{(r)} - \dots - \frac{a_{1n}}{a_{11}} x_n^{(r)} \\
 x_2^{(r+1)} &= \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}} x_1^{(r)} - \dots - \frac{a_{2n}}{a_{22}} x_n^{(r)} \\
 \vdots & \\
 x_n^{(r+1)} &= \frac{b_n}{a_{nn}} - \frac{a_{n1}}{a_{nn}} x_1^{(r)} - \dots - \frac{a_{n(n-1)}}{a_{nn}} x_{n-1}^{(r)}
 \end{aligned} \right\}$$

Briefly, we can rewrite these Equations as

$$x_i^{(r+1)} = \frac{b_i}{a_{ii}} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{a_{ij}}{a_{ii}} x_j^{(r)},$$

$$r = 1, 2, \dots, \quad i = 1, 2, \dots, n$$

It is also known as method of simultaneous displacements, since no element of $x_i^{(r+1)}$ is used in this iteration until every element is computed.

A sufficient condition for convergence of the iterative solution to the exact solution

is

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|, \quad i = 1, 2, \dots, n$$

When this condition (diagonal dominance) is true, Jacobi's method converges

Gauss–Seidel Iteration Method

**It is another well-known
iterative method for solving a
system of linear equations of
the form**

$$\left. \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{array} \right\}$$

**In Jacobi's method,
the $(r + 1)$ *th*
approximation to the
above system is given
by Equations**

$$\left. \begin{aligned}
 x_1^{(r+1)} &= \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}} x_2^{(r)} - \dots - \frac{a_{1n}}{a_{11}} x_n^{(r)} \\
 x_2^{(r+1)} &= \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}} x_1^{(r)} - \dots - \frac{a_{2n}}{a_{22}} x_n^{(r)} \\
 \vdots & \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\
 x_n^{(r+1)} &= \frac{b_n}{a_{nn}} - \frac{a_{n1}}{a_{nn}} x_1^{(r)} - \dots - \frac{a_{n(n-1)}}{a_{nn}} x_{n-1}^{(r)}
 \end{aligned} \right\}$$

**Here we can observe
that no element of $x_i^{(r+1)}$
replaces $x_i^{(r)}$ entirely
for the next cycle of
computation.**

**In Gauss-Seidel method,
the corresponding
elements of $x_i^{(r+1)}$
replaces those of $x_i^{(r)}$ as
soon as they become
available.**

Hence, it is called the method of successive displacements. For illustration consider

$$\left. \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{array} \right\}$$

**In Gauss-Seidel iteration,
the $(r + 1)$ th
approximation or
iteration is computed
from:**

$$\left. \begin{aligned}
 x_1^{(r+1)} &= \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}} x_2^{(r)} - \dots - \frac{a_{1n}}{a_{11}} x_n^{(r)} \\
 x_2^{(r+1)} &= \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}} x_1^{(r+1)} - \dots - \frac{a_{2n}}{a_{22}} x_n^{(r)} \\
 &\vdots \\
 x_n^{(r+1)} &= \frac{b_n}{a_{nn}} - \frac{a_{n1}}{a_{nn}} x_1^{(r+1)} - \dots - \frac{a_{n(n-1)}}{a_{nn}} x_{n-1}^{(r+1)}
 \end{aligned} \right\}$$

**Thus, the general procedure
can be written in the following
compact form**

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

for all $i = 1, 2, \dots, n$ and $r = 1, 2, \dots$

To describe system in the first equation, we substitute the r -th approximation into the right-hand side and denote the result by $x_1^{(r+1)}$. In the second equation, we substitute $(x_1^{(r+1)}, x_3^{(r)}, \dots, x_n^{(r)})$ and denote the result by $x_2^{(r+1)}$

In the third equation, we substitute $(x_1^{(r+1)}, x_2^{(r+1)}, x_4^{(r)}, \dots, x_n^{(r)})$ and denote the result by $x_3^{(r+1)}$, and so on. This process is continued till we arrive at the desired result. For illustration, we consider the following example :

Example

Find the solution of the following system of equations using Gauss-Seidel method and perform the first five iterations:

$$4x_1 - x_2 - x_3 = 2$$

$$-x_1 + 4x_2 - x_4 = 2$$

$$-x_1 + 4x_3 - x_4 = 1$$

$$-x_2 - x_3 + 4x_4 = 1$$

Solution

The given system of equations can be rewritten as

$$\left. \begin{aligned} x_1 &= 0.5 + 0.25x_2 + 0.25x_3 \\ x_2 &= 0.5 + 0.25x_1 + 0.25x_4 \\ x_3 &= 0.25 + 0.25x_1 + 0.25x_4 \\ x_4 &= 0.25 + 0.25x_2 + 0.25x_3 \end{aligned} \right\}$$

Taking $x_2 = x_3 = x_4 = 0$ on the right-hand side of the first equation of the system , we get $x_1^{(1)} = 0.5$. Taking $x_3 = x_4 = 0$ and the current value of x_1 , we get from the 2nd equation of the system

$$x_2^{(1)} = 0.5 + (0.25)(0.5) + 0 = 0.625$$

**Further, we take $x_4 = 0$
and the current value of x_1
we obtain from the third
equation of the system**

$$\begin{aligned}x_3^{(1)} &= 0.25 + (0.25)(0.5) + 0 \\ &= 0.375\end{aligned}$$

Now, using the current values of x_2 and x_3 the fourth equation of system gives

$$x_4^{(1)} = 0.25 + (0.25)(0.625) + (0.25)(0.375) = 0.5$$

**The Gauss-Seidel iterations
for the given set of equations
can be written as**

$$x_1^{(r+1)} = 0.5 + 0.25x_2^{(r)} + 0.25x_3^{(r)}$$

$$x_2^{(r+1)} = 0.5 + 0.25x_1^{(r+1)} + 0.25x_4^{(r)}$$

$$x_3^{(r+1)} = 0.25 + 0.25x_1^{(r+1)} + 0.25x_4^{(r)}$$

$$x_4^{(r+1)} = 0.25 + 0.25x_2^{(r+1)} + 0.25x_3^{(r+1)}$$

Now, by Gauss-Seidel procedure, the 2nd and subsequent approximations can be obtained and the sequence of the first five approximations are tabulated as below:

Variables

Iteration number r	x_1	x_2	x_3	x_4
1	0.5	0.625	0.375	0.5
2	0.75	0.8125	0.5625	0.59375
3	0.84375	0.85938	0.60938	0.61719
4	0.86719	0.87110	0.62110	0.62305
5	0.87305	0.87402	0.62402	0.62451

Numerical

Analysis

Lecture 12

Numerical

Analysis

Lecture 13

Chapter 3

Solution of Linear System of Equations and Matrix Inversion

Introduction

Gaussian Elimination

Gauss-Jordan Elimination

Crout's Reduction

Jacobi's Gauss-Seidal Iteration

Relaxation

Matrix Inversion

Gauss–Seidel Iteration Method

**It is another well-known
iterative method for solving a
system of linear equations of
the form**

$$\left. \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{array} \right\}$$

**In Jacobi's method,
the $(r + 1)$ *th*
approximation to the
above system is given
by Equations**

$$\left. \begin{aligned}
 x_1^{(r+1)} &= \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}} x_2^{(r)} - \dots - \frac{a_{1n}}{a_{11}} x_n^{(r)} \\
 x_2^{(r+1)} &= \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}} x_1^{(r)} - \dots - \frac{a_{2n}}{a_{22}} x_n^{(r)} \\
 &\vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \\
 x_n^{(r+1)} &= \frac{b_n}{a_{nn}} - \frac{a_{n1}}{a_{nn}} x_1^{(r)} - \dots - \frac{a_{n(n-1)}}{a_{nn}} x_{n-1}^{(r)}
 \end{aligned} \right\}$$

**Here we can observe
that no element of $x_i^{(r+1)}$
replaces $x_i^{(r)}$ entirely
for the next cycle of
computation.**

**In Gauss-Seidel method,
the corresponding
elements of $x_i^{(r+1)}$
replaces those of $x_i^{(r)}$ as
soon as they become
available.**

Hence, it is called the method of successive displacements. For illustration consider

$$\left. \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{array} \right\}$$

**In Gauss-Seidel iteration,
the $(r + 1)$ th
approximation or
iteration is computed
from:**

$$\left. \begin{aligned}
 x_1^{(r+1)} &= \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}} x_2^{(r)} - \dots - \frac{a_{1n}}{a_{11}} x_n^{(r)} \\
 x_2^{(r+1)} &= \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}} x_1^{(r+1)} - \dots - \frac{a_{2n}}{a_{22}} x_n^{(r)} \\
 &\vdots \\
 x_n^{(r+1)} &= \frac{b_n}{a_{nn}} - \frac{a_{n1}}{a_{nn}} x_1^{(r+1)} - \dots - \frac{a_{n(n-1)}}{a_{nn}} x_{n-1}^{(r+1)}
 \end{aligned} \right\}$$

**Thus, the general procedure
can be written in the following
compact form**

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

for all $i = 1, 2, \dots, n$ and $r = 1, 2, \dots$

To describe system in the first equation, we substitute the r -th approximation into the right-hand side and denote the result by $x_1^{(r+1)}$. In the second equation, we substitute $(x_1^{(r+1)}, x_3^{(r)}, \dots, x_n^{(r)})$ and denote the result by $x_2^{(r+1)}$

In the third equation, we substitute $(x_1^{(r+1)}, x_2^{(r+1)}, x_4^{(r)}, \dots, x_n^{(r)})$ and denote the result by $x_3^{(r+1)}$, and so on. This process is continued till we arrive at the desired result. For illustration, we consider the following example :

Relaxation Method

This is also an iterative method and is due to Southwell.

To explain the details, consider again the system of equations

$$\left. \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{array} \right\}$$

Let

$$X^{(p)} = (x_1^{(p)}, x_2^{(p)}, \dots, x_n^{(p)})^T$$

**be the solution vector
obtained iteratively after p-th
iteration. If $R_i^{(p)}$ denotes the
residual of the i-th equation
of system given above , that
is of**

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = b_i$$

defined by

$$R_i^{(p)} = b_i - a_{i1}x_1^{(p)} - a_{i2}x_2^{(p)} - \cdots - a_{in}x_n^{(p)}$$

we can improve the solution vector successively by reducing the largest residual to zero at that iteration. This is the basic idea of relaxation method.

To achieve the fast convergence of the procedure, we take all terms to one side and then reorder the equations so that the largest negative coefficients in the equations appear on the diagonal.

Now, if at any iteration, R_i is the largest residual in magnitude, then we give an increment to x_i ; a_{ii} being the coefficient of x_i

$$dx_i = \frac{R_i}{a_{ii}}$$

In other words, we change x_i .
to $(x_i + dx_i)$
to relax R_i
that is to reduce R_i to zero.

Example

Solve the system of equations

$$6x_1 - 3x_2 + x_3 = 11$$

$$2x_1 + x_2 - 8x_3 = -15$$

$$x_1 - 7x_2 + x_3 = 10$$

**by the relaxation method,
starting with the vector $(0, 0, 0)$.**

Solution

At first, we transfer all the terms to the right-hand side and reorder the equations, so that the largest coefficients in the equations appear on the diagonal.

Thus, we get

$$\left. \begin{aligned} 0 &= 11 - 6x_1 + 3x_2 - x_3 \\ 0 &= 10 - x_1 + 7x_2 - x_3 \\ 0 &= -15 - 2x_1 - x_2 + 8x_3 \end{aligned} \right\}$$

**after interchanging the 2nd
and 3rd equations.**

Starting with the initial solution vector $(0, 0, 0)$, that is taking $x_1 = x_2 = x_3 = 0$,

we find the residuals

$$R_1 = 11, \quad R_2 = 10, \quad R_3 = -15$$

of which the largest residual in magnitude is R_3 , i.e. the 3rd equation has more error and needs immediate attention for improvement.

Thus, we introduce a change, dx_3 in x_3 which is obtained from the formula

$$dx_3 = -\frac{R_3}{a_{33}} = \frac{15}{8} = 1.875$$

Similarly, we find the new residuals of large magnitude and relax it to zero, and so on.

We shall continue this process, until all the residuals are zero or very small.

Iteration		Residuals		Maximum Difference		Variables		
number	R_1	R_2	R_3	R_i	dx_i	x_1	x_2	x_3
0	11	10	-15	-15	1.875	0	0	0
1	9.125	8.125	0	9.125	1.5288	0	0	1.875
2	0.0478	6.5962	-3.0576	6.5962	-0.9423	1.5288	0	1.875

Iteration		Residuals		Maximum Difference		Variables		
number	R_1	R_2	R_3	R_i	dx_i	x_1	x_2	x_3
0	11	10	-15	-15	$15/8 = 1.875$	0	0	0
1	9.125	8.125	0	9.125	$-9.125/(-6) = 1.5288$	0	0	1.875
2	0.0478	6.5962	-3.0576	6.5962	$-6.5962/7 = -0.9423$	1.5288	0	1.875
3	-2.8747	0.0001	-2.1153	-2.8747	$2.8747/(-6) = -0.4791$	1.0497	-0.9423	1.875
4	-0.0031	0.4792	-1.1571	-1.1571	$1.1571/8 = 0.1446$	1.0497	-0.9423	1.875

Iteration		Residuals		Maximum Difference		Variables		
number	R_1	R_2	R_3	R_i	dx_i	x_1	x_2	x_3
5	-0.1447	0.3346	0.0003	0.3346	$-.3346/7$ $=-0.0478$	1.0497	-0.9423	2.0196
6	0.2881	0.0000	0.0475	0.2881	$-.2881/(-6)$ $=0.0480$	1.0497	-0.9901	2.0196
7	-0.0001	0.048	0.1435	0.1435	$=-0.0179$	1.0017	-0.9901	2.0196
8	0.0178	0.0659	0.0003	-	-	1.0017	-0.9901	2.0017

At this stage, we observe that all the residuals R_1 , R_2 and R_3 are small enough and therefore we may take the corresponding values of x_i at this iteration as the solution.

**Hence, the numerical solution
is given by**

$$x_1 = 1.0017, \quad x_2 = -0.9901, \quad x_3 = 2.0017,$$

The exact solution is

$$x_1 = 1.0, \quad x_2 = -1.0, \quad x_3 = 2.0$$

Numerical

Analysis

Lecture 13

Numerical

Analysis

Lecture 14

Chapter 3

Solution of Linear System of Equations and Matrix Inversion

Introduction

Gaussian Elimination

Gauss-Jordan Elimination

Crout's Reduction

Jacobi's

Gauss- Seidal Iteration

Relaxation

Matrix Inversion

Relaxation Method

This is also an iterative method and is due to Southwell.

To explain the details, consider again the system of equations

$$\left. \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{array} \right\}$$

Let

$$X^{(p)} = (x_1^{(p)}, x_2^{(p)}, \dots, x_n^{(p)})^T$$

**be the solution vector
obtained iteratively after p-th
iteration. If $R_i^{(p)}$ denotes the
residual of the i-th equation
of system given above , that
is of**

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = b_i$$

defined by

$$R_i^{(p)} = b_i - a_{i1}x_1^{(p)} - a_{i2}x_2^{(p)} - \cdots - a_{in}x_n^{(p)}$$

we can improve the solution vector successively by reducing the largest residual to zero at that iteration. This is the basic idea of relaxation method.

To achieve the fast convergence of the procedure, we take all terms to one side and then reorder the equations so that the largest negative coefficients in the equations appear on the diagonal.

Now, if at any iteration, R_i is the largest residual in magnitude, then we give an increment to x_i ; a_{ii} being the coefficient of x_i

$$dx_i = \frac{R_i}{a_{ii}}$$

In other words, we change x_i .
to $(x_i + dx_i)$
to relax R_i
that is to reduce R_i to zero.

Example

Solve the system of equations

$$6x_1 - 3x_2 + x_3 = 11$$

$$2x_1 + x_2 - 8x_3 = -15$$

$$x_1 - 7x_2 + x_3 = 10$$

**by the relaxation method,
starting with the vector $(0, 0, 0)$.**

Solution

At first, we transfer all the terms to the right-hand side and reorder the equations, so that the largest coefficients in the equations appear on the diagonal.

Thus, we get

$$\left. \begin{aligned} 0 &= 11 - 6x_1 + 3x_2 - x_3 \\ 0 &= 10 - x_1 + 7x_2 - x_3 \\ 0 &= -15 - 2x_1 - x_2 + 8x_3 \end{aligned} \right\}$$

**after interchanging the 2nd
and 3rd equations.**

Starting with the initial solution vector $(0, 0, 0)$, that is taking $x_1 = x_2 = x_3 = 0$,

we find the residuals

$$R_1 = 11, \quad R_2 = 10, \quad R_3 = -15$$

of which the largest residual in magnitude is R_3 , i.e. the 3rd equation has more error and needs immediate attention for improvement.

Thus, we introduce a change, dx_3 in x_3 which is obtained from the formula

$$dx_3 = -\frac{R_3}{a_{33}} = \frac{15}{8} = 1.875$$

Similarly, we find the new residuals of large magnitude and relax it to zero, and so on.

We shall continue this process, until all the residuals are zero or very small.

Iteration		Residuals		Maximum Difference		Variables		
number	R_1	R_2	R_3	R_i	dx_i	x_1	x_2	x_3
0	11	10	-15	-15	1.875	0	0	0
1	9.125	8.125	0	9.125	1.5288	0	0	1.875
2	0.0478	6.5962	-3.0576	6.5962	-0.9423	1.5288	0	1.875

Matrix Inversion

Consider a system of equations in the form

$$[A](X) = (B)$$

One way of writing its solution is in the form

$$(X) = [A]^{-1}(B)$$

Thus, the solution to the system can also be obtained if the inverse of the coefficient matrix $[A]$ is known. That is the product of two square matrices is an identity matrix

$$[A][B] = [I]$$

then, $[B] = [A]^{-1}$

and $[A] = [B]^{-1}$

Every square non-singular matrix will have an inverse.

then, $[B] = [A]^{-1}$

and $[A] = [B]^{-1}$

Every square non-singular matrix will have an inverse.

Gauss elimination and

Gauss-Jordan methods are

popular among many

methods available for finding

the inverse of a matrix.

Gaussian Elimination Method

In this method, if A is a given matrix, for which we have to find the inverse; at first, we place an identity matrix, whose order is same as that of A , adjacent to A which we call an augmented matrix.

Then the inverse of A is computed in two stages. In the first stage, A is converted into an upper triangular form, using Gaussian elimination method

In the second stage, the above upper triangular matrix is reduced to an identity matrix by row transformations. All these operations are also performed on the adjacently placed identity matrix.

Finally, when A is transformed into an identity matrix, the adjacent matrix gives the inverse of A .

In order to increase the accuracy of the result, it is essential to employ partial pivoting.

Example

Use the Gaussian elimination method to find the inverse of the matrix

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 4 & 3 & -1 \\ 3 & 5 & 3 \end{bmatrix}$$

Solution

At first, we place an identity matrix of the same order adjacent to the given matrix. Thus, the augmented matrix can be written as

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 4 & 3 & -1 & 0 & 1 & 0 \\ 3 & 5 & 3 & 0 & 0 & 1 \end{bmatrix}$$

Stage 1 (Reduction to upper triangular form): Let R_1 , R_2 and R_3 denote the 1st, 2nd and 3rd rows of a matrix. In the 1st column, 4 is the largest element, thus interchanging R_1 and R_2 to bring the pivot element 4 to the place of a_{11} , we have the augmented matrix in the form

$$\begin{bmatrix} 4 & 3 & -1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 3 & 5 & 3 & 0 & 0 & 1 \end{bmatrix}$$

Divide R_1 by 4 to get

$$\begin{bmatrix} 1 & \frac{3}{4} & -\frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 3 & 5 & 3 & 0 & 0 & 1 \end{bmatrix}$$

Perform $R_2 - R_1 \rightarrow$, which gives

$$\begin{bmatrix} 1 & \frac{3}{4} & -\frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & \frac{1}{4} & \frac{5}{4} & 1 & -\frac{1}{4} & 0 \\ 3 & 5 & 3 & 0 & 0 & 1 \end{bmatrix}$$

Perform $R_3 - 3R_1 \rightarrow R_3$ in the above equation , which yields

$$\begin{bmatrix} 1 & \frac{3}{4} & -\frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & \frac{11}{4} & \frac{15}{4} & 1 & -\frac{1}{4} & 0 \\ 0 & \frac{1}{4} & \frac{1}{4} & 0 & -\frac{3}{4} & 1 \end{bmatrix}$$

Now, looking at the second column for the pivot, the $\max(1/4, 11/4)$ is $11/4$.

Therefore, we interchange R_2 and R_3 in the last equation and get

$$\begin{bmatrix} 1 & \frac{3}{4} & -\frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & \frac{11}{4} & \frac{15}{4} & 0 & -\frac{3}{4} & 1 \\ 0 & \frac{1}{4} & \frac{5}{4} & 1 & -\frac{1}{4} & 0 \end{bmatrix}$$

**Now, divide R_2 by the pivot a_{22}
= 11/4, and obtain**

$$\begin{bmatrix} 1 & \frac{3}{4} & -\frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & 1 & \frac{15}{11} & 0 & -\frac{3}{11} & \frac{4}{11} \\ 0 & \frac{1}{4} & \frac{5}{4} & 1 & -\frac{1}{4} & 0 \end{bmatrix}$$

Performing $R_3 - (1/4)R_2 \rightarrow R_3$
yields

$$\begin{bmatrix} 1 & \frac{3}{4} & -\frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & 1 & \frac{15}{11} & 0 & -\frac{3}{11} & \frac{4}{11} \\ 0 & 0 & \frac{10}{11} & 1 & -\frac{2}{11} & -\frac{1}{11} \end{bmatrix}$$

**Finally, we divide R_3 by $(10/11)$,
thus getting an upper
triangular form**

$$\begin{bmatrix} 1 & \frac{3}{4} & -\frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & 1 & \frac{15}{11} & 0 & -\frac{3}{11} & \frac{4}{11} \\ 0 & 0 & 1 & \frac{11}{10} & -\frac{1}{5} & -\frac{1}{10} \end{bmatrix}$$

Stage II

Reduction to an identity matrix

$(1/4)R_3 + R_1$ and $(-15/11)R_3 + R_2$

$$\begin{bmatrix} 1 & \frac{3}{4} & 0 & \frac{11}{40} & \frac{1}{5} & -\frac{1}{40} \\ 0 & 1 & 0 & -\frac{3}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 1 & \frac{11}{10} & -\frac{1}{5} & -\frac{1}{10} \end{bmatrix}$$

**Finally, performing $R_1 \leftarrow R_1 - (3/4)R_2$
we obtain**

$$\begin{bmatrix} 1 & 0 & 0 & \frac{7}{5} & \frac{1}{5} & -\frac{2}{5} \\ 0 & 1 & 0 & -\frac{3}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 1 & \frac{11}{10} & -\frac{1}{5} & -\frac{1}{10} \end{bmatrix}$$

Thus, we have

$$A^{-1} = \begin{bmatrix} \frac{7}{5} & \frac{1}{5} & -\frac{2}{5} \\ -\frac{3}{2} & 0 & \frac{1}{2} \\ \frac{11}{10} & -\frac{1}{5} & -\frac{1}{10} \end{bmatrix}$$

Gauss - Jordan Method

This method is similar to Gaussian elimination method, with the essential difference that the stage I of reducing the given matrix to an upper triangular form is not needed.

However, the given matrix can be directly reduced to an identity matrix using elementary row operations.

Example

Find the inverse of the given matrix by Gauss-Jordan method

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 4 & 3 & -1 \\ 3 & 5 & 3 \end{bmatrix}$$

Solution

Let R_1 , R_2 and R_3 denote the 1st, 2nd and 3rd rows of a matrix. We place the identity matrix adjacent to the given matrix. So the augmented matrix is given by

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 4 & 3 & -1 & 0 & 1 & 0 \\ 3 & 5 & 3 & 0 & 0 & 1 \end{bmatrix}$$

Performing $R_2 - 4R_1 \rightarrow R_2$, we get

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & -1 & -5 & 0 & 1 & 0 \\ 3 & 5 & 3 & 0 & 0 & 1 \end{bmatrix}$$

**Now, performing $R_3 - 3R_1 \rightarrow R_3$,
we obtain**

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & -1 & -5 & -4 & 1 & 0 \\ 0 & 2 & 0 & -3 & 0 & 1 \end{bmatrix}$$

Carrying out further operations

$$R_2 + R_1 \rightarrow R_1 \qquad R_3 \text{ and } R_2 \rightarrow R_3,$$

we arrive at

$$\begin{bmatrix} 1 & 0 & -4 & -3 & 1 & 0 \\ 0 & -1 & -5 & -4 & 1 & 0 \\ 0 & 0 & -10 & -11 & 2 & 1 \end{bmatrix}$$

**Now, dividing the third row by
−10, we get**

$$\begin{bmatrix} 1 & 0 & -4 & -3 & 1 & 0 \\ 0 & -1 & -5 & -4 & 1 & 0 \\ 0 & 0 & 1 & \frac{11}{10} & -1 & -\frac{1}{10} \end{bmatrix}$$

Further, we perform $R_1 + 4R_3 \rightarrow R_1$,
and $R_2 + 5R_3 \rightarrow R_2$ **to get**

$$\begin{bmatrix} 1 & 0 & 0 & \frac{7}{5} & \frac{1}{5} & -\frac{2}{5} \\ 0 & -1 & 0 & \frac{3}{2} & 0 & -\frac{1}{2} \\ 0 & 0 & 1 & \frac{11}{10} & -\frac{1}{5} & -\frac{1}{10} \end{bmatrix}$$

Finally, multiplying R_2 by -1 , we obtain

$$\begin{bmatrix} 1 & 0 & 0 & \frac{7}{5} & \frac{1}{5} & -\frac{2}{5} \\ 0 & 1 & 0 & -\frac{3}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 1 & \frac{11}{10} & -\frac{1}{5} & -\frac{1}{10} \end{bmatrix}$$

Hence, we have

$$A^{-1} = \begin{bmatrix} \frac{7}{5} & \frac{1}{5} & -\frac{2}{5} \\ -\frac{3}{2} & 0 & \frac{1}{2} \\ \frac{11}{10} & -\frac{1}{5} & -\frac{1}{10} \end{bmatrix}$$

Numerical

Analysis

Lecture 14

Numerical

Analysis

Lecture 15

Chapter 4

Eigen Value Problems

**Let $[A]$ be an $n \times n$ square matrix.
Suppose, there exists a scalar λ
and a vector**

$$X = (x_1 \quad x_2 \quad \dots \quad x_n)^T$$

such that

$$[A](X) = \lambda(X)$$

$$\frac{d}{dx}(e^{ax}) = a(e^{ax})$$

$$\frac{d^2}{dx^2}(\sin ax) = -a^2(\sin ax)$$

**Then λ is the eigen value
and X is the corresponding
eigenvector of the matrix $[A]$.**

We can also write it as

$$[A - \lambda I](X) = (O)$$

This represents a set of n homogeneous equations possessing non-trivial solution, provided

$$|A - \lambda I| = 0$$

This determinant, on expansion, gives an n -th degree polynomial which is called characteristic polynomial of $[A]$, which has n roots. Corresponding to each root, we can solve these equations in principle, and determine a vector called eigenvector.

Finding the roots of the characteristic equation is laborious. Hence, we look for better methods suitable from the point of view of computation. Depending upon the type of matrix $[A]$ and on what one is looking for, various numerical methods are available.

Power Method

Jacobi's Method

Note!

**We shall consider
only real and real-
symmetric matrices
and discuss power
and Jacobi's methods**

Power Method

To compute the largest eigen value and the corresponding eigenvector of the system

$$[A](X) = \lambda(X)$$

where $[A]$ is a real, symmetric or un-symmetric matrix, the power method is widely used in practice.

Procedure

Step 1: Choose the initial vector such that the largest element is unity.

Step 2: The normalized vector $v^{(0)}$ is pre-multiplied by the matrix $[A]$.

Step 3: The resultant vector is again normalized.

Step 4: This process of iteration is continued and the new normalized vector is repeatedly pre-multiplied by the matrix $[A]$ until the required accuracy is obtained.

At this point, the result looks like

$$u^{(k)} = [A]v^{(k-1)} = q_k v^{(k)}$$

Here, q_k is the desired largest eigen value and $v^{(k)}$ is the corresponding eigenvector.

Example

Find the eigen value of largest modulus, and the associated eigenvector of the matrix by power method

$$[A] = \begin{bmatrix} 2 & 3 & 2 \\ 4 & 3 & 5 \\ 3 & 2 & 9 \end{bmatrix}$$

Solution

We choose an initial vector $v^{(0)}$ as $(1,1,1)^T$.

Then, compute first iteration

$$u^{(1)} = [A]v^{(0)} = \begin{bmatrix} 2 & 3 & 2 \\ 4 & 3 & 5 \\ 3 & 2 & 9 \end{bmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 7 \\ 12 \\ 14 \end{pmatrix}$$

Now we normalize the resultant vector to get

$$u^{(1)} = 14 \begin{pmatrix} \frac{1}{2} \\ \frac{6}{7} \\ 1 \end{pmatrix} = q_1 v^{(1)}$$

The second iteration gives,

$$u^{(2)} = [A]v^{(1)} \begin{bmatrix} 2 & 3 & 2 \\ 4 & 3 & 5 \\ 3 & 2 & 9 \end{bmatrix} \begin{pmatrix} \frac{1}{2} \\ \frac{6}{7} \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{39}{7} \\ \frac{67}{7} \\ \frac{171}{14} \end{pmatrix}$$

$$= 12.2143 \begin{pmatrix} 0.456140 \\ 0.783626 \\ 1.0 \end{pmatrix} = q_2 v^{(2)}$$

**Continuing this
procedure, the third
and subsequent
iterations are given in
the following slides**

$$u^{(3)} = [A]v^{(2)} = \begin{bmatrix} 2 & 3 & 2 \\ 4 & 3 & 5 \\ 3 & 2 & 9 \end{bmatrix} \begin{pmatrix} 0.456140 \\ 0.783626 \\ 1.0 \end{pmatrix}$$

$$= \begin{pmatrix} 5.263158 \\ 9.175438 \\ 11.935672 \end{pmatrix} = 11.935672 \begin{pmatrix} 0.44096 \\ 0.776874 \\ 1.0 \end{pmatrix} = q_3 v^{(3)}$$

$$\begin{aligned} u^{(4)} &= [A]v^{(3)} = \begin{pmatrix} 5.18814 \\ 9.07006 \\ 11.86036 \end{pmatrix} \\ &= 11.8636 \begin{pmatrix} 0.437435 \\ 0.764737 \\ 1.0 \end{pmatrix} = q_4 v^{(4)} \end{aligned}$$

$$\begin{aligned} u^{(5)} &= [A]v^{(4)} = \begin{pmatrix} 5.16908 \\ 9.04395 \\ 11.84178 \end{pmatrix} \\ &= 11.84178 \begin{pmatrix} 0.436512 \\ 0.763732 \\ 1.0 \end{pmatrix} = q_5 v^{(5)} \end{aligned}$$

After rounding-off, the largest eigen value and the corresponding eigenvector as accurate to two decimals are

$$\lambda = 11.84$$

$$(X) = \begin{pmatrix} 0.44 \\ 0.76 \\ 1.00 \end{pmatrix}$$

Exercise

Find the largest eigen value and the corresponding eigen vector by power method after fourth iteration starting with the initial vector

$$v^{(0)} = (0, 0, 1)^T$$

$$[A] = \begin{bmatrix} 1 & -3 & 2 \\ 4 & 4 & -1 \\ 6 & 3 & 5 \end{bmatrix}$$

Let

$$\lambda_1, \lambda_2, \dots, \lambda_n$$

be the distinct eigen values of an $n \times n$ matrix $[A]$, such that

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$$

and suppose

$$v_1, v_2, \dots, v_n$$

are the corresponding eigen vectors

Power method is applicable if the above eigen values are real and distinct, and hence, the corresponding eigenvectors are linearly independent.

Then, any eigenvector v in the space spanned by the eigenvectors

$$v_1, v_2, \dots, v_n$$

can be written as their linear combination

$$v = c_1 v_1 + c_2 v_2 + \dots + c_n v_n$$

Pre-multiplying by A and substituting

$$Av_1 = \lambda_1 v_1, \quad Av_2 = \lambda_2 v_2, \quad \dots \quad Av_n = \lambda_n v_n$$

We get

$$Av = \lambda_1 \left(c_1 v_1 + c_2 \frac{\lambda_2}{\lambda_1} v_2 + \dots + c_n \frac{\lambda_n}{\lambda_1} v_n \right)$$

Again, pre-multiplying by A and simplifying, we obtain

$$A^2 v = \lambda_1^2 \left[c_1 v_1 + c_2 \left(\frac{\lambda_2}{\lambda_1} \right)^2 v_2 + \cdots + c_n \left(\frac{\lambda_n}{\lambda_1} \right)^2 v_n \right]$$

Similarly, we have

$$A^r v = \lambda_1^r \left[c_1 v_1 + c_2 \left(\frac{\lambda_2}{\lambda_1} \right)^r v_2 + \cdots + c_n \left(\frac{\lambda_n}{\lambda_1} \right)^r v_n \right]$$

and

$$A^{r+1} v = (\lambda_1)^{r+1} \left[c_1 v_1 + c_2 \left(\frac{\lambda_2}{\lambda_1} \right)^{r+1} v_2 + \cdots + c_n \left(\frac{\lambda_n}{\lambda_1} \right)^{r+1} v_n \right]$$

Now, the eigen value

$$\lambda_1$$

can be computed as the limit of the ratio of the corresponding components of

$$A^r v$$

and

$$A^{r+1} v.$$

That is,

$$\lambda_1 = \frac{\lambda_1^{r+1}}{\lambda_1^r} = \lim_{r \rightarrow \infty} \frac{(A^{r+1} v)_p}{(A^r v)_p}, \quad p = 1, 2, \dots, n$$

Here, the index p stands for the p -th component in the corresponding vector

Sometimes, we may be interested in finding the least eigen value and the corresponding eigenvector.

In that case, we proceed as follows.

We note that $[A](X) = \lambda(X)$.

Pre-multiplying by $[A^{-1}]$, we get

$$[A^{-1}][A](X) = [A^{-1}]\lambda(X) = \lambda[A^{-1}](X)$$

**Which can be rewritten
as**

$$[A^{-1}](X) = \frac{1}{\lambda}(X)$$

**which shows that the
inverse matrix has a set
of eigen values which are
the reciprocals of the
eigen values of $[A]$.**

Thus, for finding the eigen value of the least magnitude of the matrix $[A]$, we have to apply power method to the inverse of $[A]$.

Numerical

Analysis

Lecture 15

Numerical

Analysis

Lecture 16

Chapter 4

Eigen Value Problems

**Let $[A]$ be an $n \times n$ square matrix.
Suppose, there exists a scalar λ
and a vector**

$$X = (x_1 \quad x_2 \quad \dots \quad x_n)^T$$

such that

$$[A](X) = \lambda(X)$$

**Then λ is the eigen value
and X is the corresponding
eigenvector of the matrix $[A]$.**

We can also write it as

$$[A - \lambda I](X) = (O)$$

This represents a set of n homogeneous equations possessing non-trivial solution, provided

$$|A - \lambda I| = 0$$

This determinant, on expansion, gives an n -th degree polynomial which is called characteristic polynomial of $[A]$, which has n roots. Corresponding to each root, we can solve these equations in principle, and determine a vector called eigenvector.

Finding the roots of the characteristic equation is laborious. Hence, we look for better methods suitable from the point of view of computation. Depending upon the type of matrix $[A]$ and on what one is looking for, various numerical methods are available.

Power Method

Jacobi's Method

Note!

**We shall consider
only real and real-
symmetric matrices
and discuss power
and Jacobi's methods**

Power Method

To compute the largest eigen value and the corresponding eigenvector of the system

$$[A](X) = \lambda(X)$$

where $[A]$ is a real, symmetric or un-symmetric matrix, the power method is widely used in practice.

Procedure

Step 1: Choose the initial vector such that the largest element is unity.

Step 2: The normalized vector $v^{(0)}$ is pre-multiplied by the matrix $[A]$.

Step 3: The resultant vector is again normalized.

Step 4: This process of iteration is continued and the new normalized vector is repeatedly pre-multiplied by the matrix $[A]$ until the required accuracy is obtained.

Let

$$\lambda_1, \lambda_2, \dots, \lambda_n$$

be the distinct eigen values of an $n \times n$ matrix $[A]$, such that

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$$

and suppose

$$v_1, v_2, \dots, v_n$$

are the corresponding eigen vectors

Power method is applicable if the above eigen values are real and distinct, and hence, the corresponding eigenvectors are linearly independent.

Then, any eigenvector v in the space spanned by the eigenvectors

$$v_1, v_2, \dots, v_n$$

can be written as their linear combination

$$v = c_1 v_1 + c_2 v_2 + \dots + c_n v_n$$

Pre-multiplying by A and substituting

$$Av_1 = \lambda_1 v_1, \quad Av_2 = \lambda_2 v_2, \quad \dots \quad Av_n = \lambda_n v_n$$

We get

$$Av = \lambda_1 \left(c_1 v_1 + c_2 \frac{\lambda_2}{\lambda_1} v_2 + \dots + c_n \frac{\lambda_n}{\lambda_1} v_n \right)$$

Again, pre-multiplying by A and simplifying, we obtain

$$A^2 v = \lambda_1^2 \left[c_1 v_1 + c_2 \left(\frac{\lambda_2}{\lambda_1} \right)^2 v_2 + \cdots + c_n \left(\frac{\lambda_n}{\lambda_1} \right)^2 v_n \right]$$

Similarly, we have

$$A^r v = \lambda_1^r \left[c_1 v_1 + c_2 \left(\frac{\lambda_2}{\lambda_1} \right)^r v_2 + \cdots + c_n \left(\frac{\lambda_n}{\lambda_1} \right)^r v_n \right]$$

and

$$A^{r+1} v = (\lambda_1)^{r+1} \left[c_1 v_1 + c_2 \left(\frac{\lambda_2}{\lambda_1} \right)^{r+1} v_2 + \cdots + c_n \left(\frac{\lambda_n}{\lambda_1} \right)^{r+1} v_n \right]$$

Now, the eigen value

$$\lambda_1$$

can be computed as the limit of the ratio of the corresponding components of

$$A^r v$$

and

$$A^{r+1} v.$$

That is,

$$\lambda_1 = \frac{\lambda_1^{r+1}}{\lambda_1^r} = \lim_{r \rightarrow \infty} \frac{(A^{r+1} v)_p}{(A^r v)_p}, \quad p = 1, 2, \dots, n$$

Here, the index p stands for the p -th component in the corresponding vector

Sometimes, we may be interested in finding the least eigen value and the corresponding eigenvector.

In that case, we proceed as follows.

We note that $[A](X) = \lambda(X)$.

Pre-multiplying by $[A^{-1}]$, we get

$$[A^{-1}][A](X) = [A^{-1}]\lambda(X) = \lambda[A^{-1}](X)$$

**Which can be rewritten
as**

$$[A^{-1}](X) = \frac{1}{\lambda}(X)$$

**which shows that the
inverse matrix has a set
of eigen values which are
the reciprocals of the
eigen values of $[A]$.**

Thus, for finding the eigen value of the least magnitude of the matrix $[A]$, we have to apply power method to the inverse of $[A]$.

Jacobi's Method

Definition

An $n \times n$ matrix $[A]$ is said to be orthogonal if

$$[A]^T [A] = [I],$$

$$\text{i.e. } [A]^T = [A]^{-1}$$

In order to compute all the eigen values and the corresponding eigenvectors of a real symmetric matrix, Jacobi's method is highly recommended. It is based on an important property from matrix theory, which states

that, if $[A]$ is an $n \times n$ real symmetric matrix, its eigen values are real, and there exists an orthogonal matrix $[S]$ such that the diagonal matrix D is

$$[S^{-1}][A][S]$$

This diagonalization can be carried out by applying a series of orthogonal transformations

$$S_1, S_2, \dots, S_n,$$

Let A be an $n \times n$ real symmetric matrix. Suppose $|a_{ij}|$ be numerically the largest element amongst the off-diagonal elements of A . We construct an orthogonal matrix S_1 defined as

$$s_{ij} = -\sin \theta, \quad s_{ji} = \sin \theta,$$

$$s_{ii} = \cos \theta, \quad s_{jj} = \cos \theta$$

While each of the remaining off-diagonal elements are zero, the remaining diagonal elements are assumed to be unity. Thus, we construct S_1 as under

$$\begin{array}{ccccccc}
 & & & \text{i-th column} & & \text{j-th column} & & \\
 & & & \downarrow & & \downarrow & & \\
 S_1 = & \left[\begin{array}{ccccccc}
 1 & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\
 0 & 1 & \dots & 0 & \dots & 0 & \dots & 0 \\
 \vdots & \vdots & & \vdots & & \vdots & & \vdots \\
 0 & 0 & \dots & \cos \theta & \dots & -\sin \theta & \dots & 0 & \leftarrow \text{i-th row} \\
 \vdots & \vdots & & \vdots & & \vdots & & \vdots \\
 0 & 0 & \dots & \sin \theta & \dots & \cos \theta & \dots & 0 & \leftarrow \text{j-th row} \\
 \vdots & \vdots & & \vdots & & \vdots & & \vdots \\
 0 & 0 & \dots & 0 & \dots & 0 & \dots & 1
 \end{array} \right]
 \end{array}$$

where $\cos \theta, -\sin \theta, \sin \theta$ and $\cos \vartheta$

are inserted in $(i, i), (i, j), (j, i), (j, j)$ – th positions respectively, and elsewhere it is identical with a unit matrix.

Now, we compute

$$D_1 = S_1^{-1} A S_1 = S_1^T A S_1$$

Since S_1 is an orthogonal matrix, such that $S_1^{-1} = S_1^T$.

After the transformation, the elements at the position (i, j) , (j, i) get annihilated, that is d_{ij} and d_{ji} reduce to zero, which is seen as follows:

$$\begin{bmatrix} d_{ii} & d_{ij} \\ d_{ji} & d_{jj} \end{bmatrix}$$

$$= \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} a$$

$$\begin{bmatrix} a_{ii} \cos^2 \theta \sin \theta \cos \theta + a_{jj} \sin^2 \theta & (a_{jj} - a_{ii}) \sin \theta \cos \theta + a_{ij} \cos 2\theta \\ (a_{ji} - a_{ii}) \sin \theta \cos \theta + a_{ij} \cos 2\theta & a_{ii} \sin^2 \theta + a_{jj} \cos^2 \theta - 2a_{ij} \sin \theta \cos \theta \end{bmatrix}$$

Therefore, $d_{ij} = 0$, only if,

$$a_{ij} \cos 2\theta + \frac{a_{jj} - a_{ii}}{2} \sin 2\theta = 0$$

That is if

$$\tan 2\theta = \frac{2a_{ij}}{a_{ii} - a_{jj}}$$

Thus, we choose θ such that the above equation is satisfied, thereby, the pair of off-diagonal elements d_{ij} and d_{ji} reduces to zero.

However, though it creates a new pair of zeros, it also introduces non-zero contributions at formerly zero positions.

Also, the above equation gives four values of θ , but to get the least possible rotation, we choose

$$-\pi/4 \leq \theta \leq \pi/4.$$

As a next step, the numerically largest off-diagonal element in the newly obtained rotated matrix D_1 is identified and the above procedure is repeated using another orthogonal matrix S_2 to get D_2 . That is we obtain

$$D_2 = S_2^{-1} D_1 S_2 = S_2^T (S_1^T A S_1) S_2$$

Similarly, we perform a series of such two-dimensional rotations or orthogonal transformations. After making r transformations, we obtain

$$\begin{aligned} D_r &= S_r^{-1} S_{r-1}^{-1} \dots S_2^{-1} S_1^{-1} A S_1 S_2 \dots S_{r-1} S_r \\ &= (S_1 S_2 \dots S_{r-1} S_r)^{-1} A (S_1 S_2 \dots S_{r-1} S_r) \\ &= S^{-1} A S \end{aligned}$$

Now, as $r \rightarrow \infty$, D_r approaches to a diagonal matrix, with the eigen values on the main diagonal. The corresponding eigenvectors are the columns of S .

It is estimated that the minimum number of rotations required to transform the given $n \times n$ real symmetric matrix $[A]$ into a diagonal form is $n(n-1)/2$

Numerical

Analysis

Lecture 16

Numerical

Analysis

Lecture 17

Chapter 4

Eigen Value Problems

**Let $[A]$ be an $n \times n$ square matrix.
Suppose, there exists a scalar λ
and a vector**

$$X = (x_1 \quad x_2 \quad \dots \quad x_n)^T$$

such that

$$[A](X) = \lambda(X)$$

Power Method

Jacobi's Method

Power Method

To compute the largest eigen value and the corresponding eigenvector of the system

$$[A](X) = \lambda(X)$$

where $[A]$ is a real, symmetric or un-symmetric matrix, the power method is widely used in practice.

Procedure

Step 1: Choose the initial vector such that the largest element is unity.

Step 2: The normalized vector $v^{(0)}$ is pre-multiplied by the matrix $[A]$.

Step 3: The resultant vector is again normalized.

Step 4: This process of iteration is continued and the new normalized vector is repeatedly pre-multiplied by the matrix $[A]$ until the required accuracy is obtained.

We calculate

$$A^r v = \lambda_1^r \left[c_1 v_1 + c_2 \left(\frac{\lambda_2}{\lambda_1} \right)^r v_2 + \cdots + c_n \left(\frac{\lambda_n}{\lambda_1} \right)^r v_n \right]$$

$$A^{r+1} v = (\lambda_1)^{r+1} \left[c_1 v_1 + c_2 \left(\frac{\lambda_2}{\lambda_1} \right)^{r+1} v_2 + \cdots + c_n \left(\frac{\lambda_n}{\lambda_1} \right)^{r+1} v_n \right]$$

Now, the eigen value

$$\lambda_1$$

can be computed as the limit of the ratio of the corresponding components of

$$A^r v$$

and

$$A^{r+1} v.$$

That is,

$$\lambda_1 = \frac{\lambda_1^{r+1}}{\lambda_1^r} = \lim_{r \rightarrow \infty} \frac{(A^{r+1} v)_p}{(A^r v)_p}, \quad p = 1, 2, \dots, n$$

Here, the index p stands for the p -th component in the corresponding vector

Sometimes, we may be interested in finding the least eigen value and the corresponding eigenvector.

In that case, we proceed as follows.

We note that $[A](X) = \lambda(X)$.

Pre-multiplying by $[A^{-1}]$, we get

$$[A^{-1}][A](X) = [A^{-1}]\lambda(X) = \lambda[A^{-1}](X)$$

$$[A^{-1}](X) = \frac{1}{\lambda}(X)$$

The inverse matrix has a set of eigen values which are the reciprocals of the eigen values of $[A]$.

Thus, for finding the eigen value of the least magnitude of the matrix $[A]$, we have to apply power method to the inverse of $[A]$.

Jacobi's Method

Definition

An $n \times n$ matrix $[A]$ is said to be orthogonal if

$$[A]^T [A] = [I],$$

$$\text{i.e. } [A]^T = [A]^{-1}$$

If $[A]$ is an $n \times n$ real symmetric matrix, its eigen values are real, and there exists an orthogonal matrix $[S]$ such that the diagonal matrix D is

$$[S^{-1}][A][S]$$

This diagonalization can be carried out by applying a series of orthogonal transformations

$$S_1, S_2, \dots, S_n,$$

Let A be an $n \times n$ real symmetric matrix. Suppose $|a_{ij}|$ be numerically the largest element amongst the off-diagonal elements of A . We construct an orthogonal matrix S_1 defined as

$$s_{ij} = -\sin \theta, \quad s_{ji} = \sin \theta,$$

$$s_{ii} = \cos \theta, \quad s_{jj} = \cos \theta$$

$$\begin{array}{ccccccc}
 & & & \text{i-th column} & & \text{j-th column} & & \\
 & & & \downarrow & & \downarrow & & \\
 S_1 = & \begin{bmatrix}
 1 & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\
 0 & 1 & \dots & 0 & \dots & 0 & \dots & 0 \\
 \vdots & \vdots & & \vdots & & \vdots & & \vdots \\
 0 & 0 & \dots & \cos \theta & \dots & -\sin \theta & \dots & 0 \\
 \vdots & \vdots & & \vdots & & \vdots & & \vdots \\
 0 & 0 & \dots & \sin \theta & \dots & \cos \theta & \dots & 0 \\
 \vdots & \vdots & & \vdots & & \vdots & & \vdots \\
 0 & 0 & \dots & 0 & \dots & 0 & \dots & 1
 \end{bmatrix}
 & \begin{array}{l}
 \leftarrow \text{i-th row} \\
 \\
 \leftarrow \text{j-th row}
 \end{array}
 \end{array}$$

where $\cos \theta, -\sin \theta, \sin \theta$ and $\cos \vartheta$

are inserted in $(i, i), (i, j), (j, i), (j, j)$ – th positions respectively, and elsewhere it is identical with a unit matrix.

Now, we compute

$$D_1 = S_1^{-1} A S_1 = S_1^T A S_1$$

Therefore, $d_{ij} = 0$, only if,

$$a_{ij} \cos 2\theta + \frac{a_{jj} - a_{ii}}{2} \sin 2\theta = 0$$

That is if

$$\tan 2\theta = \frac{2a_{ij}}{a_{ii} - a_{jj}}$$

Thus, we choose θ such that the above equation is satisfied, thereby, the pair of off-diagonal elements d_{ij} and d_{ji} reduces to zero.

However, though it creates a new pair of zeros, it also introduces non-zero contributions at formerly zero positions.

Also, the above equation gives four values of θ , but to get the least possible rotation, we choose

$$-\pi/4 \leq \theta \leq \pi/4.$$

As a next step, the numerically largest off-diagonal element in the newly obtained rotated matrix D_1 is identified and the above procedure is repeated using another orthogonal matrix S_2 to get D_2 . That is we obtain

$$D_2 = S_2^{-1} D_1 S_2 = S_2^T (S_1^T A S_1) S_2$$

Similarly, we perform a series of such two-dimensional rotations or orthogonal transformations. After making r transformations, we obtain

$$\begin{aligned} D_r &= S_r^{-1} S_{r-1}^{-1} \dots S_2^{-1} S_1^{-1} A S_1 S_2 \dots S_{r-1} S_r \\ &= (S_1 S_2 \dots S_{r-1} S_r)^{-1} A (S_1 S_2 \dots S_{r-1} S_r) \\ &= S^{-1} A S \end{aligned}$$

Example

Find all the eigen values and the corresponding eigen vectors of the matrix by Jacobi's method

$$A = \begin{bmatrix} 1 & \sqrt{2} & 2 \\ \sqrt{2} & 3 & \sqrt{2} \\ 2 & \sqrt{2} & 1 \end{bmatrix}$$

Solution

The given matrix is real and symmetric. The largest off-diagonal element is found to be

$$a_{13} = a_{31} = 2.$$

Now, we compute

$$\tan 2\theta = \frac{2a_{ij}}{a_{ii} - a_{jj}} = \frac{2a_{13}}{a_{11} - a_{33}} = \frac{4}{0} = \infty$$

Which gives, $\theta = \pi/4$

Thus, we construct an orthogonal matrix S_i as

$$S_1 = \begin{bmatrix} \cos \frac{\pi}{4} & 0 & -\sin \frac{\pi}{4} \\ 0 & 1 & 0 \\ \sin \frac{\pi}{4} & 0 & \cos \frac{\pi}{4} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}$$

The first rotation gives,

$$D_1 = S_1^{-1} A S_1$$

$$= \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & \sqrt{2} & 2 \\ \sqrt{2} & 3 & \sqrt{2} \\ 2 & \sqrt{2} & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}$$

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

We observe that the elements d_{13} and d_{31} got annihilated. To make sure that calculations are correct up to this step, we see that the sum of the diagonal elements of D_1 is same as the sum of the diagonal elements of the original matrix A .

As a second step, we choose the largest off-diagonal element of D_1 and is found to be $d_{12} = d_{21} = 2$, and compute

$$\tan 2\theta = \frac{2d_{12}}{d_{11} - d_{22}} = \frac{4}{0} = \infty$$

which again gives $\theta = \pi/4$

Thus, we construct the second rotation matrix as

$$S_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

At the end of the second rotation, we get

$$D_2 = S_2^{-1} D_1 S_2$$

$$= \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 2 & 0 \\ 2 & 3 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Which turned out to be a diagonal matrix, so we stop the computation. From here, we notice that the eigen values of the given matrix are 5, 1 and -1. The eigenvectors are the column vectors of

$$S = S_1 S_2$$

Therefore

$$S = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

Example

Find all the eigen values of the matrix by Jacobi's method.

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

Solution

Here all the off-diagonal elements are of the same order of magnitude. Therefore, we can choose any one of them.

Suppose, we choose a_{12} as the largest element and compute

$$\tan 2\theta = \frac{-1}{0} = \infty$$

Which gives, $\theta = \pi/4$.

**Then $\cos \theta = \sin \theta = 1/\sqrt{2}$
and we construct an orthogonal
matrix S , such that**

$$S = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The first rotation gives

$$D_1 = S_1^{-1} A S_1$$

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

$$= \begin{bmatrix} 1 & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 3 & -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 2 \end{bmatrix}$$

Now, we choose $d_{13} = -1/\sqrt{2}$

**as the largest element of D_1 and
compute**

$$\tan 2\theta = \frac{2d_{13}}{d_{11} - d_{33}} = \frac{-\sqrt{2}}{1-2}$$

$$\theta = 27^\circ 22' 41''.$$

Now we construct another orthogonal matrix S_2 , such that

$$S_2 = \begin{bmatrix} 0.888 & 0 & -0.459 \\ 0 & 1 & 0 \\ 0.459 & 0 & 0.888 \end{bmatrix}$$

At the end of second rotation, we obtain

$$D_2 = S_2^{-1} D_1 S_2 = \begin{bmatrix} 0.634 & -0.325 & 0 \\ 0.325 & 3 & -0.628 \\ 0 & -0.628 & 2.365 \end{bmatrix}$$

Now, the numerically largest off-diagonal element of D_2 is found to be $d_{23} = -0.628$ and compute

$$\tan 2\theta = \frac{-2 \times 0.628}{3 - 2.365}$$

$$\theta = -31^{\circ}35'24''.$$

Thus, the orthogonal matrix is

$$S_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.852 & 0.524 \\ 0 & -0.524 & 0.852 \end{bmatrix}$$

**At the end of third rotation,
we get**

$$D_3 = S_3^{-1} D_2 S_3 = \begin{bmatrix} 0.634 & -0.277 & 0 \\ 0.277 & 3.386 & 0 \\ 0 & 0 & 1.979 \end{bmatrix}$$

**To reduce D_3 to a diagonal form,
some more rotations are required.
However, we may take 0.634, 3.386
and 1.979 as eigen values of the
given matrix.**

Numerical

Analysis

Lecture 17

Numerical

Analysis

Lecture 18

Chapter 5

Interpolation

Finite Difference Operators
Newton's Forward Difference
Interpolation Formula
Newton's Backward Difference
Interpolation Formula
Lagrange's Interpolation
Formula
Divided Differences
Interpolation in Two Dimensions
Cubic Spline Interpolation

Introduction

Finite differences play an important role in numerical techniques, where tabulated values of the functions are available. For instance, consider a function

$$y = f(x).$$

As x takes values

$$x_0, x_1, x_2, \dots, x_n,$$

**let the corresponding values
of y be**

$$y_0, y_1, y_2, \dots, y_n.$$

That is, for given a table of values,

$$(x_k, y_k), k = 0, 1, 2, \dots, n;$$

the process of estimating the value of y , for any intermediate value of x , is called interpolation.

The method of computing the value of y , for a given value of x , lying outside the table of values of x is known as extrapolation.

If the function $f(x)$ is known, the value of y corresponding to any x can be readily computed to the desired accuracy.

For interpolation of a tabulated function, the concept of finite differences is important. The knowledge about various finite difference operators and their symbolic relations are very much needed to establish various interpolation formulae.

Finite Difference Operators

Finite Difference Operators

Forward Differences

Backward Differences

Central Difference

Forward Differences

For a given table of values

$$(x_k, y_k), k = 0, 1, 2, \dots, n$$

**with equally spaced abscissas
of a function**

$$y = f(x),$$

**we define the forward difference
operator Δ as follows**

$$\Delta y_i = y_{i+1} - y_i, \quad i = 0, 1, \dots, (n-1)$$

To be explicit, we write

$$\Delta y_0 = y_1 - y_0$$

$$\Delta y_1 = y_2 - y_1$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots$$

$$\Delta y_{n-1} = y_n - y_{n-1}$$

These differences are called *first differences of the function y* and are denoted by the symbol Δy_i

Here Δ is called the first difference operator

Similarly, the differences of the first differences are called second differences, defined by

$$\Delta^2 y_0 = \Delta y_1 - \Delta y_0, \quad \Delta^2 y_1 = \Delta y_2 - \Delta y_1$$

Thus, in general

$$\Delta^2 y_i = \Delta y_{i+1} - \Delta y_i$$

Here Δ^2 is called the *second difference operator*. Thus, continuing, we can define, *r-th* difference of y , as

$$\Delta^r y_i = \Delta^{r-1} y_{i+1} - \Delta^{r-1} y_i$$

By defining a difference table as a convenient device for displaying various differences, the above defined differences can be written down systematically by constructing a difference table for values

$$(x_k, y_k), k = 0, 1, \dots, 6$$

Forward Difference Table

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$	$\Delta^6 y$
x_0	y_0	Δy_0					
x_1	y_1	Δy_1	$\Delta^2 y_0$	$\Delta^3 y_0$	$\Delta^4 y_0$		
x_2	y_2	Δy_2	$\Delta^2 y_1$	$\Delta^3 y_1$	$\Delta^4 y_1$	$\Delta^5 y_0$	
x_3	y_3	Δy_3	$\Delta^2 y_2$	$\Delta^3 y_2$	$\Delta^4 y_1$	$\Delta^5 y_1$	$\Delta^6 y_0$
x_4	y_4	Δy_4	$\Delta^2 y_3$	$\Delta^3 y_3$			
x_5	y_5	Δy_5	$\Delta^2 y_4$				
x_6	y_6						

This difference table is called *forward difference* table or *diagonal difference* table.

Here, each difference is located in its appropriate column, midway between the elements of the previous column.

Please note that the subscript remains constant along each diagonal of the table. The first term in the table, that is y_0 is called the leading term, while the differences

$$\Delta y_0, \Delta^2 y_0, \Delta^3 y_0, \dots$$

are called leading differences

Example

Construct a forward difference table for the following values of x and y :

x	0.1	0.3	0.5	0.7	0.9	1.1	1.3
y	0.003	0.067	0.148	0.248	0.37	0.518	0.697

Solution

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$
0.1	0.003					
0.3	0.067	0.064	0.017			
0.5	0.148	0.081	0.019	0.002		
0.7	0.248	0.1	0.022	0.003	0.001	
0.9	0.37	0.122	0.026	0.004	0.001	0
1.1	0.518	0.148	0.031	0.005	0.001	0
1.3	0.697	0.179				

Example

Express $\Delta^2 y_0$ and $\Delta^3 y_0$ in terms of the values of the function y .

Solution:

Noting that each higher order difference is defined in terms of the lower order difference, we have

$$\begin{aligned}\Delta^2 y_0 &= \Delta y_1 - \Delta y_0 = (y_2 - y_1) - (y_1 - y_0) \\ &= y_2 - 2y_1 + y_0\end{aligned}$$

and

$$\begin{aligned}\Delta^3 y_0 &= \Delta^2 y_1 - \Delta^2 y_0 = (\Delta y_2 - \Delta y_1) - (\Delta y_1 - \Delta y_0) \\ &= (y_3 - y_2) - (y_2 - y_1) - (y_2 - y_1) + (y_1 - y_0) \\ &= y_3 - 3y_2 + 3y_1 - y_0\end{aligned}$$

Hence, we observe that the coefficients of the values of y , in the expansion of $\Delta^2 y_0, \Delta^3 y_0$, are binomial coefficients.

Thus, in general, we arrive at the following result: -

$$\Delta^n y_0 = y_n - {}^nC_1 y_{n-1} + {}^nC_2 y_{n-2} - {}^nC_3 y_{n-3} + \cdots + (-1)^n y_0$$

Example

Show that the value of y_n can be expressed in terms of the leading value y_0 and the leading differences

$$\Delta y_0, \Delta^2 y_0, \dots, \Delta^n y_0.$$

Solution

The forward difference table will be

$$\left. \begin{array}{l} y_1 - y_0 = \Delta y_0 \quad \text{or} \quad y_1 = y_0 + \Delta y_0 \\ y_2 - y_1 = \Delta y_1 \quad \text{or} \quad y_2 = y_1 + \Delta y_1 \\ y_3 - y_2 = \Delta y_2 \quad \text{or} \quad y_3 = y_2 + \Delta y_2 \end{array} \right\}$$

Similarly,

$$\left. \begin{array}{l} \Delta y_1 - \Delta y_0 = \Delta^2 y_0 \quad \text{or} \quad \Delta y_1 = \Delta y_0 + \Delta^2 y_0 \\ \Delta y_2 - \Delta y_1 = \Delta^2 y_1 \quad \text{or} \quad \Delta y_2 = \Delta y_1 + \Delta^2 y_1 \end{array} \right\}$$

Similarly, we can also write

$$\left. \begin{array}{l} \Delta^2 y_1 - \Delta^2 y_0 = \Delta^3 y_0 \quad \text{or} \quad \Delta^2 y_1 = \Delta^2 y_0 + \Delta^3 y_0 \\ \Delta^2 y_2 - \Delta^2 y_1 = \Delta^3 y_1 \quad \text{or} \quad \Delta^2 y_2 = \Delta^2 y_1 + \Delta^3 y_1 \end{array} \right\}$$

$$\begin{aligned}\Delta y_2 &= (\Delta y_0 + \Delta^2 y_0) + (\Delta^2 y_0 + \Delta^3 y_0) \\ &= \Delta y_0 + 2\Delta^2 y_0 + \Delta^3 y_0\end{aligned}$$

$$\begin{aligned}y_3 &= y_2 + \Delta y_2 \\ &= (y_1 + \Delta y_1) + (\Delta y_1 + \Delta^2 y_1) \\ &= y_0 + 3\Delta y_0 + 3\Delta^2 y_0 + \Delta^3 y_0 \\ &= (1 + \Delta)^3 y_0\end{aligned}$$

Similarly, we can symbolically write

$$y_1 = (1 + \Delta)y_0,$$

$$y_2 = (1 + \Delta)^2 y_0,$$

$$y_3 = (1 + \Delta)^3 y_0$$

.....

$$y_n = (1 + \Delta)^n y_0$$

Hence, we obtain

$$y_n = y_0 + {}^nC_1 \Delta y_0 + {}^nC_2 \Delta^2 y_0 \\ + {}^nC_3 \Delta^3 y_0 + \cdots + \Delta^n y_0$$

$$y_n = \sum_{i=0}^n {}^nC_i \Delta^i y_0$$

Numerical

Analysis

Lecture 18

Backward Differences

For a given table of values

$$(x_k, y_k), k = 0, 1, 2, \dots, n$$

**of a function $y = f(x)$ with
equally spaced abscissas, the
first backward differences are
usually expressed in terms of
the backward difference
operator ∇ as**

$$\nabla y_i = y_i - y_{i-1} \quad i = n, (n-1), \dots, 1$$

OR

$$\nabla y_1 = y_1 - y_0$$

$$\nabla y_2 = y_2 - y_1$$

$$\vdots \qquad \qquad \vdots \qquad \qquad \vdots$$

$$\nabla y_n = y_n - y_{n-1}$$

The differences of these differences are called *second differences* and they are denoted by $\nabla^2 y_2, \nabla^2 y_3, \dots, \nabla^2 y_n$.

That is

$$\nabla^2 y_1 = \nabla y_2 - \nabla y_1$$

$$\nabla^2 y_2 = \nabla y_3 - \nabla y_2$$

$$\vdots \qquad \qquad \vdots \qquad \qquad \vdots$$

$$\nabla^2 y_n = \nabla y_n - \nabla y_{n-1}$$

Thus, in general, the second backward differences are

$$\nabla^2 y_i = \nabla y_i - \nabla y_{i-1}, \quad i = n, (n-1), \dots, 2$$

while the *k-th* backward differences are given as

$$\nabla^k y_i = \nabla^{k-1} y_i - \nabla^{k-1} y_{i-1}, \quad i = n, (n-1), \dots, k$$

**These backward differences
can be systematically
arranged for a table of values**

$$(x_k, y_k), k = 0, 1, \dots, 6$$

Backward Difference Table

x	y	∇y	$\nabla^2 y$	$\nabla^3 y$	$\nabla^4 y$	$\nabla^5 y$	$\nabla^6 y$
x_0	y_0						
x_1	y_1	∇y_1	$\nabla^2 y_2$	$\nabla^3 y_3$	$\nabla^4 y_4$		
x_2	y_2	∇y_2	$\nabla^2 y_3$	$\nabla^3 y_4$	$\nabla^4 y_5$	$\nabla^5 y_5$	
x_3	y_3	∇y_3	$\nabla^2 y_4$	$\nabla^3 y_5$	$\nabla^4 y_6$	$\nabla^5 y_6$	$\nabla^6 y_6$
x_4	y_4	∇y_4	$\nabla^2 y_5$	$\nabla^3 y_6$			
x_5	y_5	∇y_5	$\nabla^2 y_6$				
x_6	y_6	∇y_6					

From this table, it can be observed that the subscript remains constant along every backward diagonal.

Example

Show that any value of y can be expressed in terms of y_n and its backward differences.

Solution

From $\nabla y_i = y_i - y_{i-1} \quad i = n, (n-1), \dots, 1$

We get $y_{n-1} = y_n - \nabla y_n$ $y_{n-2} = y_{n-1} - \nabla y_{n-1}$

From $\nabla^2 y_i = \nabla y_i - \nabla y_{i-1}, \quad i = n, (n-1), \dots, 2$

We get $\nabla y_{n-1} = \nabla y_n - \nabla^2 y_n$

From these equations, we obtain

$$y_{n-2} = y_n - 2\nabla y_n + \nabla^2 y_n$$

Similarly, we can show that

$$y_{n-3} = y_n - 3\nabla y_n + 3\nabla^2 y_n - \nabla^3 y_n$$

Symbolically, these results can be rewritten as follows:

$$y_{n-1} = (1 - \nabla) y_n$$

$$y_{n-2} = (1 - \nabla)^2 y_n$$

$$y_{n-3} = (1 - \nabla)^3 y_n$$

.....

$$y_{n-r} = (1 - \nabla)^r y_n$$

$$y_{n-r} = y_n - {}^nC_1 \nabla y_n + {}^nC_2 \nabla^2 y_n - \cdots + (-1)^r \nabla^r y_n$$

Central Differences

In some applications, central difference notation is found to be more convenient to represent the successive differences of a function. Here, we use the symbol δ to represent central difference operator and the subscript of δy for any difference as the average of the subscripts

$$\delta y_{1/2} = y_1 - y_0, \quad \delta y_{3/2} = y_2 - y_1,$$

In general

$$\delta y_i = y_{i+(1/2)} - y_{i-(1/2)}$$

Higher order differences are defined as follows:

$$\delta^2 y_i = \delta y_{i+(1/2)} - \delta y_{i-(1/2)}$$

$$\delta^n y_i = \delta^{n-1} y_{i+(1/2)} - \delta^{n-1} y_{i-(1/2)}$$

These central differences can be systematically arranged as indicated in the Table

x	y	δy	$\delta^2 y$	$\delta^3 y$	$\delta^4 y$	$\delta^5 y$	$\delta^6 y$
x_0	y_0						
x_1	y_1	$\delta y_{1/2}$	$\delta^2 y_1$	$\delta^3 y_{3/2}$	$\delta^4 y_2$		
x_2	y_2	$\delta y_{3/2}$	$\delta^2 y_2$	$\delta^3 y_{5/2}$	$\delta^4 y_3$	$\delta^5 y_{5/2}$	
x_3	y_3	$\delta y_{5/2}$	$\delta^2 y_3$	$\delta^3 y_{7/2}$	$\delta^4 y_4$	$\delta^5 y_{7/2}$	$\delta^6 y_3$
x_4	y_4	$\delta y_{7/2}$	$\delta^2 y_4$	$\delta^3 y_{9/2}$			
x_5	y_5	$\delta y_{9/2}$	$\delta^2 y_5$				
x_6	y_6	$\delta y_{11/2}$					

Thus, we observe that all the odd differences have a fractional suffix and all the even differences with the same subscript lie horizontally.

The following alternative notation may also be adopted to introduce finite difference operators. Let $y = f(x)$ be a functional relation between x and y , which is also denoted by y_x .

Suppose, we are given consecutive values of x differing by h say $x, x + h, x + 2h, x + 3h$, etc. The corresponding values of y are $y_x, y_{x+h}, y_{x+2h}, y_{x+3h}$,

As before, we can form the differences of these values.

Thus

$$\Delta y_x = y_{x+h} - y_x = f(x+h) - f(x)$$

$$\Delta^2 y_x = \Delta y_{x+h} - \Delta y_x$$

Similarly

$$\nabla y_x = y_x - y_{x-h} = f(x) - f(x-h)$$

$$\delta y_x = y_{x+(h/2)} - y_{x-(h/2)} = f\left(x + \frac{h}{2}\right) - f\left(x - \frac{h}{2}\right)$$

Numerical

Analysis

Lecture 18

Numerical

Analysis

Lecture 19

Chapter 5

Interpolation

Finite Difference Operators
Newton's Forward Difference
Interpolation Formula
Newton's Backward Difference
Interpolation Formula
Lagrange's Interpolation
Formula
Divided Differences
Interpolation in Two Dimensions
Cubic Spline Interpolation

Introduction

Finite differences play an important role in numerical techniques, where tabulated values of the functions are available. For instance, consider a function

$$y = f(x).$$

As x takes values

$$x_0, x_1, x_2, \dots, x_n,$$

**let the corresponding values
of y be**

$$y_0, y_1, y_2, \dots, y_n.$$

That is, for given a table of values,

$$(x_k, y_k), k = 0, 1, 2, \dots, n;$$

the process of estimating the value of y , for any intermediate value of x , is called interpolation.

The method of computing the value of y , for a given value of x , lying outside the table of values of x is known as extrapolation.

If the function $f(x)$ is known, the value of y corresponding to any x can be readily computed to the desired accuracy.

For interpolation of a tabulated function, the concept of finite differences is important. The knowledge about various finite difference operators and their symbolic relations are very much needed to establish various interpolation formulae.

Finite Difference Operators

Finite Difference Operators

Forward Differences

Backward Differences

Central Difference

Forward Differences

For a given table of values

$$(x_k, y_k), k = 0, 1, 2, \dots, n$$

**with equally spaced abscissas
of a function**

$$y = f(x),$$

**we define the forward difference
operator Δ as follows**

$$\Delta y_i = y_{i+1} - y_i, \quad i = 0, 1, \dots, (n-1)$$

To be explicit, we write

$$\Delta y_0 = y_1 - y_0$$

$$\Delta y_1 = y_2 - y_1$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots$$

$$\Delta y_{n-1} = y_n - y_{n-1}$$

These differences are called *first differences of the function y* and are denoted by the symbol Δy_i

Here Δ is called the first difference operator

Similarly, the differences of the first differences are called second differences, defined by

$$\Delta^2 y_0 = \Delta y_1 - \Delta y_0, \quad \Delta^2 y_1 = \Delta y_2 - \Delta y_1$$

Thus, in general

$$\Delta^2 y_i = \Delta y_{i+1} - \Delta y_i$$

Here Δ^2 is called the *second difference operator*. Thus, continuing, we can define, *r-th* difference of y , as

$$\Delta^r y_i = \Delta^{r-1} y_{i+1} - \Delta^{r-1} y_i$$

By defining a difference table as a convenient device for displaying various differences, the above defined differences can be written down systematically by constructing a difference table for values

$$(x_k, y_k), k = 0, 1, \dots, 6$$

Forward Difference Table

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$	$\Delta^6 y$
x_0	y_0	Δy_0					
x_1	y_1	Δy_1	$\Delta^2 y_0$	$\Delta^3 y_0$	$\Delta^4 y_0$		
x_2	y_2	Δy_2	$\Delta^2 y_1$	$\Delta^3 y_1$	$\Delta^4 y_1$	$\Delta^5 y_0$	
x_3	y_3	Δy_3	$\Delta^2 y_2$	$\Delta^3 y_2$	$\Delta^4 y_1$	$\Delta^5 y_1$	$\Delta^6 y_0$
x_4	y_4	Δy_4	$\Delta^2 y_3$	$\Delta^3 y_3$			
x_5	y_5	Δy_5	$\Delta^2 y_4$				
x_6	y_6						

Backward Differences

For a given table of values

$$(x_k, y_k), k = 0, 1, 2, \dots, n$$

**of a function $y = f(x)$ with
equally spaced abscissas, the
first backward differences are
usually expressed in terms of
the backward difference
operator ∇ as**

$$\nabla y_i = y_i - y_{i-1} \quad i = n, (n-1), \dots, 1$$

OR

$$\nabla y_1 = y_1 - y_0$$

$$\nabla y_2 = y_2 - y_1$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots$$

$$\nabla y_n = y_n - y_{n-1}$$

The differences of these differences are called *second differences* and they are denoted by $\nabla^2 y_2, \nabla^2 y_3, \dots, \nabla^2 y_n$.

That is

$$\nabla^2 y_1 = \nabla y_2 - \nabla y_1$$

$$\nabla^2 y_2 = \nabla y_3 - \nabla y_2$$

$$\vdots \qquad \qquad \vdots \qquad \qquad \vdots$$

$$\nabla^2 y_n = \nabla y_n - \nabla y_{n-1}$$

Thus, in general, the second backward differences are

$$\nabla^2 y_i = \nabla y_i - \nabla y_{i-1}, \quad i = n, (n-1), \dots, 2$$

while the *k*-th backward differences are given as

$$\nabla^k y_i = \nabla^{k-1} y_i - \nabla^{k-1} y_{i-1}, \quad i = n, (n-1), \dots, k$$

**These backward differences
can be systematically
arranged for a table of values**

$$(x_k, y_k), k = 0, 1, \dots, 6$$

Backward Difference Table

x	y	∇y	$\nabla^2 y$	$\nabla^3 y$	$\nabla^4 y$	$\nabla^5 y$	$\nabla^6 y$
x_0	y_0						
x_1	y_1	∇y_1	$\nabla^2 y_2$	$\nabla^3 y_3$	$\nabla^4 y_4$		
x_2	y_2	∇y_2	$\nabla^2 y_3$	$\nabla^3 y_4$	$\nabla^4 y_5$	$\nabla^5 y_5$	
x_3	y_3	∇y_3	$\nabla^2 y_4$	$\nabla^3 y_5$	$\nabla^4 y_6$	$\nabla^5 y_6$	$\nabla^6 y_6$
x_4	y_4	∇y_4	$\nabla^2 y_5$	$\nabla^3 y_6$			
x_5	y_5	∇y_5	$\nabla^2 y_6$				
x_6	y_6	∇y_6					

From this table, it can be observed that the subscript remains constant along every backward diagonal.

Example

Show that any value of y can be expressed in terms of y_n and its backward differences.

Solution

From $\nabla y_i = y_i - y_{i-1} \quad i = n, (n-1), \dots, 1$

We get $y_{n-1} = y_n - \nabla y_n$ $y_{n-2} = y_{n-1} - \nabla y_{n-1}$

From $\nabla^2 y_i = \nabla y_i - \nabla y_{i-1}, \quad i = n, (n-1), \dots, 2$

We get $\nabla y_{n-1} = \nabla y_n - \nabla^2 y_n$

From these equations, we obtain

$$y_{n-2} = y_n - 2\nabla y_n + \nabla^2 y_n$$

Similarly, we can show that

$$y_{n-3} = y_n - 3\nabla y_n + 3\nabla^2 y_n - \nabla^3 y_n$$

Symbolically, these results can be rewritten as follows:

$$y_{n-1} = (1 - \nabla) y_n$$

$$y_{n-2} = (1 - \nabla)^2 y_n$$

$$y_{n-3} = (1 - \nabla)^3 y_n$$

.....

$$y_{n-r} = (1 - \nabla)^r y_n$$

$$y_{n-r} = y_n - {}^nC_1 \nabla y_n + {}^nC_2 \nabla^2 y_n - \cdots + (-1)^r \nabla^r y_n$$

Central Differences

In some applications, central difference notation is found to be more convenient to represent the successive differences of a function. Here, we use the symbol δ to represent central difference operator and the subscript of δy for any difference as the average of the subscripts

$$\delta y_{1/2} = y_1 - y_0, \quad \delta y_{3/2} = y_2 - y_1,$$

In general

$$\delta y_i = y_{i+(1/2)} - y_{i-(1/2)}$$

Higher order differences are defined as follows:

$$\delta^2 y_i = \delta y_{i+(1/2)} - \delta y_{i-(1/2)}$$

$$\delta^n y_i = \delta^{n-1} y_{i+(1/2)} - \delta^{n-1} y_{i-(1/2)}$$

These central differences can be systematically arranged as indicated in the Table

x	y	δy	$\delta^2 y$	$\delta^3 y$	$\delta^4 y$	$\delta^5 y$	$\delta^6 y$
x_0	y_0						
x_1	y_1	$\delta y_{1/2}$	$\delta^2 y_1$	$\delta^3 y_{3/2}$	$\delta^4 y_2$		
x_2	y_2	$\delta y_{3/2}$	$\delta^2 y_2$	$\delta^3 y_{5/2}$	$\delta^4 y_3$	$\delta^5 y_{5/2}$	
x_3	y_3	$\delta y_{5/2}$	$\delta^2 y_3$	$\delta^3 y_{7/2}$	$\delta^4 y_4$	$\delta^5 y_{7/2}$	$\delta^6 y_3$
x_4	y_4	$\delta y_{7/2}$	$\delta^2 y_4$	$\delta^3 y_{9/2}$			
x_5	y_5	$\delta y_{9/2}$	$\delta^2 y_5$				
x_6	y_6	$\delta y_{11/2}$					

Thus, we observe that all the odd differences have a fractional suffix and all the even differences with the same subscript lie horizontally.

The following alternative notation may also be adopted to introduce finite difference operators. Let $y = f(x)$ be a functional relation between x and y , which is also denoted by y_x .

Suppose, we are given consecutive values of x differing by h say $x, x + h, x + 2h, x + 3h$, etc. The corresponding values of y are $y_x, y_{x+h}, y_{x+2h}, y_{x+3h}$,

As before, we can form the differences of these values.

Thus

$$\Delta y_x = y_{x+h} - y_x = f(x+h) - f(x)$$

$$\Delta^2 y_x = \Delta y_{x+h} - \Delta y_x$$

Similarly

$$\nabla y_x = y_x - y_{x-h} = f(x) - f(x-h)$$

$$\delta y_x = y_{x+(h/2)} - y_{x-(h/2)} = f\left(x + \frac{h}{2}\right) - f\left(x - \frac{h}{2}\right)$$

$$\Delta y_i = y_{i+1} - y_i, \quad i = 0, 1, \dots, (n-1)$$

To be explicit, we write

$$\Delta y_0 = y_1 - y_0$$

$$\Delta y_1 = y_2 - y_1$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots$$

$$\Delta y_{n-1} = y_n - y_{n-1}$$

$$\nabla y_i = y_i - y_{i-1} \quad i = n, (n-1), \dots, 1$$

OR

$$\nabla y_1 = y_1 - y_0$$

$$\nabla y_2 = y_2 - y_1$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots$$

$$\nabla y_n = y_n - y_{n-1}$$

$$\delta y_{1/2} = y_1 - y_0, \quad \delta y_{3/2} = y_2 - y_1,$$

In general

$$\delta y_i = y_{i+(1/2)} - y_{i-(1/2)}$$

Higher order differences are defined as follows:

$$\delta^2 y_i = \delta y_{i+(1/2)} - \delta y_{i-(1/2)}$$

$$\delta^n y_i = \delta^{n-1} y_{i+(1/2)} - \delta^{n-1} y_{i-(1/2)}$$

Numerical

Analysis

Lecture 19

Numerical

Analysis

Lecture 20

Chapter 5

Interpolation

Finite Difference Operators
Newton's Forward Difference
Interpolation Formula
Newton's Backward Difference
Interpolation Formula
Lagrange's Interpolation
Formula
Divided Differences
Interpolation in Two Dimensions
Cubic Spline Interpolation

Introduction

Finite differences play an important role in numerical techniques, where tabulated values of the functions are available. For instance, consider a function

$$y = f(x).$$

As x takes values

$$x_0, x_1, x_2, \dots, x_n,$$

**let the corresponding values
of y be**

$$y_0, y_1, y_2, \dots, y_n.$$

That is, for given a table of values,

$$(x_k, y_k), k = 0, 1, 2, \dots, n;$$

the process of estimating the value of y , for any intermediate value of x , is called interpolation.

The method of computing the value of y , for a given value of x , lying outside the table of values of x is known as extrapolation.

Finite Difference Operators

Forward Differences

Backward Differences

Central Difference

$$\Delta y_i = y_{i+1} - y_i, \quad i = 0, 1, \dots, (n-1)$$

To be explicit, we write

$$\Delta y_0 = y_1 - y_0$$

$$\Delta y_1 = y_2 - y_1$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots$$

$$\Delta y_{n-1} = y_n - y_{n-1}$$

Similarly, the differences of the first differences are called second differences, defined by

$$\Delta^2 y_0 = \Delta y_1 - \Delta y_0, \quad \Delta^2 y_1 = \Delta y_2 - \Delta y_1$$

Thus, in general

$$\Delta^2 y_i = \Delta y_{i+1} - \Delta y_i$$

Here Δ^2 is called the *second difference operator*. Thus, continuing, we can define, *r-th* difference of y , as

$$\Delta^r y_i = \Delta^{r-1} y_{i+1} - \Delta^{r-1} y_i$$

Backward Differences

$$\nabla y_i = y_i - y_{i-1} \quad i = n, (n-1), \dots, 1$$

OR

$$\nabla y_1 = y_1 - y_0$$

$$\nabla y_2 = y_2 - y_1$$

$$\vdots \qquad \qquad \vdots \qquad \qquad \vdots$$

$$\nabla y_n = y_n - y_{n-1}$$

The differences of these differences are called *second differences* and they are denoted by $\nabla^2 y_2, \nabla^2 y_3, \dots, \nabla^2 y_n$.

That is

$$\nabla^2 y_1 = \nabla y_2 - \nabla y_1$$

$$\nabla^2 y_2 = \nabla y_3 - \nabla y_2$$

$$\vdots \qquad \qquad \vdots \qquad \qquad \vdots$$

$$\nabla^2 y_n = \nabla y_n - \nabla y_{n-1}$$

Thus, in general, the second backward differences are

$$\nabla^2 y_i = \nabla y_i - \nabla y_{i-1}, \quad i = n, (n-1), \dots, 2$$

while the *k-th* backward differences are given as

$$\nabla^k y_i = \nabla^{k-1} y_i - \nabla^{k-1} y_{i-1}, \quad i = n, (n-1), \dots, k$$

$$\delta y_{1/2} = y_1 - y_0, \quad \delta y_{3/2} = y_2 - y_1,$$

In general

$$\delta y_i = y_{i+(1/2)} - y_{i-(1/2)}$$

Higher order differences are defined as follows:

$$\delta^2 y_i = \delta y_{i+(1/2)} - \delta y_{i-(1/2)}$$

$$\delta^n y_i = \delta^{n-1} y_{i+(1/2)} - \delta^{n-1} y_{i-(1/2)}$$

Thus

$$\Delta y_x = y_{x+h} - y_x = f(x+h) - f(x)$$

$$\Delta^2 y_x = \Delta y_{x+h} - \Delta y_x$$

Similarly

$$\nabla y_x = y_x - y_{x-h} = f(x) - f(x-h)$$

$$\delta y_x = y_{x+(h/2)} - y_{x-(h/2)} = f\left(x + \frac{h}{2}\right) - f\left(x - \frac{h}{2}\right)$$

Shift operator, E

Let $y = f(x)$ be a function of x , and let x takes the consecutive values $x, x + h, x + 2h$, etc. We then define an operator having the property

$$E f(x) = f(x + h)$$

Thus, when E operates on $f(x)$, the result is the next value of the function. Here, E is called the shift operator. If we apply the operator E twice on $f(x)$, we get

$$\begin{aligned} E^2 f(x) &= E[E f(x)] \\ &= E[f(x+h)] = f(x+2h) \end{aligned}$$

Thus, in general, if we apply the operator ' E ' n times on $f(x)$, we get

$$E^n f(x) = f(x + nh)$$

or

$$E^n y_x = y_{x+nh}$$

$$Ey_0 = y_1, \quad E^2 y_0 = y_2, \quad E^4 y_0 = y_4, \quad \dots, \quad E^2 y_2 = y_4$$

**The inverse operator E^{-1}
is defined as**

$$E^{-1} f(x) = f(x - h)$$

Similarly,

$$E^{-n} f(x) = f(x - nh)$$

Average Operator, μ

$$\begin{aligned}\mu f(x) &= \frac{1}{2} \left[f\left(x + \frac{h}{2}\right) + f\left(x - \frac{h}{2}\right) \right] \\ &= \frac{1}{2} \left[y_{x+(h/2)} + y_{x-(h/2)} \right]\end{aligned}$$

Differential Operator, D

$$Df(x) = \frac{d}{dx} f(x) = f'(x)$$

$$D^2 f(x) = \frac{d^2}{dx^2} f(x) = f''(x)$$

Important Results Using

$\Delta, \nabla, \delta, E, \mu$

$$\begin{aligned}\Delta y_x &= y_{x+h} - y_x = Ey_x - y_x \\ &= (E - 1)y_x\end{aligned}$$

$$\Rightarrow \Delta = E - 1$$

$$\begin{aligned}\nabla y_x &= y_x - y_{x-h} = y_x - E^{-1}y_x \\ &= (1 - E^{-1})y_x\end{aligned}$$

$$\Rightarrow \nabla = 1 - E^{-1} = \frac{E - 1}{E}$$

$$\begin{aligned}\delta y_x &= y_{x+(h/2)} - y_{x-(h/2)} \\ &= E^{1/2} y_x - E^{-1/2} y_x \\ &= (E^{1/2} - E^{-1/2}) y_x\end{aligned}$$

$$\delta = E^{1/2} - E^{-1/2}$$

The definition of μ and E similarly yields

$$\begin{aligned}\mu y_x &= \frac{1}{2} \left[y_{x+(h/2)} + y_{x-(h/2)} \right] \\ &= \frac{1}{2} (E^{1/2} + E^{-1/2}) y_x\end{aligned}$$

$$\Rightarrow \mu = \frac{1}{2} (E^{1/2} + E^{-1/2})$$

We know that

$$Ey_x = y_{x+h} = f(x+h)$$

$$Ey_x = f(x) + hf'(x) + \frac{h^2}{2!} f''(x) + \cdots$$

$$= f(x) + hDf(x) + \frac{h^2}{2!} D^2 f(x) + \cdots$$

$$= \left(1 + \frac{hD}{1!} + \frac{h^2 D^2}{2!} + \cdots \right) f(x) = e^{hD} y_x$$

Thus, $hD = \log E$

Example

Prove that

$$hD = \log(1 + \Delta)$$

$$= -\log(1 - \nabla)$$

$$= \sinh^{-1}(\mu\delta)$$

Solution

Using the standard relations we have

$$hD = \log E$$

$$= \log(1 + \Delta)$$

$$= -\log E^{-1}$$

$$= -\log(1 - \nabla)$$

$$\mu\delta = \frac{1}{2}(E^{1/2} + E^{-1/2})(E^{1/2} - E^{-1/2})$$

$$= \frac{1}{2}(E - E^{-1})$$

$$= \frac{1}{2}(e^{hD} - e^{-hD})$$

$$= \sinh(hD)$$

$$\Rightarrow hD = \sinh^{-1} \mu\delta$$

Example Prove that

(i)

$$1 + \delta^2 \mu^2 = \left(1 + \frac{\delta^2}{2}\right)^2$$

(ii)

$$E^{1/2} = \mu + \frac{\delta}{2}$$

(iii)

$$\Delta = \frac{\delta^2}{2} + \delta \sqrt{1 + (\delta^2 / 4)}$$

Cont! Prove that

(iv)

$$\mu\delta = \frac{\Delta E^{-1}}{2} + \frac{\Delta}{2}$$

(v)

$$\mu\delta = \frac{\Delta + \nabla}{2}$$

Solution

From the definition, we have:

$$\mu\delta = \frac{1}{2}(E^{1/2} + E^{-1/2})(E^{1/2} - E^{-1/2}) = \frac{1}{2}(E - E^{-1})$$

$$\therefore 1 + \mu^2\delta^2 = 1 + \frac{1}{4}(E^2 - 2 + E^{-2}) = \frac{1}{4}(E + E^{-1})^2$$

$$1 + \frac{\delta^2}{2} = 1 + \frac{1}{2}(E^{1/2} - E^{-1/2})^2 = \frac{1}{2}(E + E^{-1})^2$$

(ii) Now

$$\mu + (\delta / 2)$$

$$= \frac{1}{2} (E^{1/2} + E^{-1/2} + E^{1/2} - E^{-1/2}) = E^{1/2}$$

Thus, the second result is proved.

(iii) We can write

$$\frac{\delta^2}{2} + \delta \sqrt{1 + (\delta^2 / 4)} = \frac{(E^{1/2} - E^{-1/2})^2}{2} + \frac{(E^{1/2} - E^{-1/2}) \sqrt{1 + \frac{1}{4}(E^{1/2} - E^{-1/2})^2}}{1}$$

$$= \frac{E - 2 + E^{-1}}{2} + \frac{1}{2}(E^{1/2} - E^{-1/2})(E^{1/2} + E^{-1/2})$$

$$= \frac{E - 2 + E^{-1}}{2} + \frac{E - E^{-1}}{2}$$

$$= E - 1 = \Delta$$

(iv) We have

$$\mu\delta = \frac{1}{2}(E^{1/2} + E^{-1/2})(E^{1/2} - E^{-1/2}) = \frac{1}{2}(E - E^{-1})$$

$$= \frac{1}{2}(1 + \Delta - E^{-1}) = \frac{\Delta}{2} + \frac{1}{2}(1 - E^{-1})$$

$$= \frac{\Delta}{2} + \frac{1}{2}\left(\frac{E-1}{E}\right) = \frac{\Delta}{2} + \frac{\Delta}{2E}$$

(v) We can write

$$\begin{aligned}\mu\delta &= \frac{1}{2}(E^{1/2} + E^{-1/2})(E^{1/2} - E^{-1/2}) \\ &= \frac{1}{2}(E - E^{-1})\end{aligned}$$

$$= \frac{1}{2}(1 + \Delta - 1 + \nabla) = \frac{1}{2}(\Delta + \nabla)$$

Example

Show that the operations μ and E commute.

$$E\mu = \mu E$$

Numerical

Analysis

Lecture 20

Numerical

Analysis

Lecture 21

Chapter 5

Interpolation

Finite Difference Operators
Newton's Forward Difference
Interpolation Formula
Newton's Backward Difference
Interpolation Formula
Lagrange's Interpolation
Formula
Divided Differences
Interpolation in Two Dimensions
Cubic Spline Interpolation

For given a table of values,

$$(x_k, y_k), k = 0, 1, 2, \dots, n;$$

**the process of estimating
the value of y , for any
intermediate value of x , is
called interpolation.**

Method of computing the value of y , for a given value of x , lying outside the table of values of x is known as extrapolation.

$$\Delta^r y_i = \Delta^{r-1} y_{i+1} - \Delta^{r-1} y_i$$

$$\nabla^k y_i = \nabla^{k-1} y_i - \nabla^{k-1} y_{i-1},$$

$$i = n, (n-1), \dots, k$$

$$\delta^n y_i = \delta^{n-1} y_{i+(1/2)} - \delta^{n-1} y_{i-(1/2)}$$

Thus

$$\Delta y_x = y_{x+h} - y_x = f(x+h) - f(x)$$

$$\Delta^2 y_x = \Delta y_{x+h} - \Delta y_x$$

Similarly

$$\nabla y_x = y_x - y_{x-h} = f(x) - f(x-h)$$

$$\delta y_x = y_{x+(h/2)} - y_{x-(h/2)} = f\left(x + \frac{h}{2}\right) - f\left(x - \frac{h}{2}\right)$$

Shift operator, E

$$E f(x) = f(x + h)$$

$$E^n f(x) = f(x + nh)$$

$$E^n y_x = y_{x+nh}$$

**The inverse operator E^{-1}
is defined as**

$$E^{-1} f(x) = f(x - h)$$

Similarly,

$$E^{-n} f(x) = f(x - nh)$$

Average Operator, μ

$$\begin{aligned}\mu f(x) &= \frac{1}{2} \left[f\left(x + \frac{h}{2}\right) + f\left(x - \frac{h}{2}\right) \right] \\ &= \frac{1}{2} \left[y_{x+(h/2)} + y_{x-(h/2)} \right]\end{aligned}$$

Differential Operator, D

$$Df(x) = \frac{d}{dx} f(x) = f'(x)$$

$$D^2 f(x) = \frac{d^2}{dx^2} f(x) = f''(x)$$

Important Results

$$\Delta = E - 1$$

$$\nabla = 1 - E^{-1} = \frac{E - 1}{E}$$

$$\delta = E^{1/2} - E^{-1/2}$$

$$hD = \log E$$

$$\mu = \frac{1}{2}(E^{1/2} + E^{-1/2})$$

Newton's Forward Difference Interpolation Formula

Let $y = f(x)$ be a function which takes values $f(x_0)$, $f(x_0 + h)$, $f(x_0 + 2h)$, ..., corresponding to various equi-spaced values of x with spacing h , say x_0 , $x_0 + h$, $x_0 + 2h$, ...

Suppose, we wish to evaluate the function $f(x)$ for a value $x_0 + ph$, where p is any real number, then for any real number p , we have the operator E such that

$$E^p f(x) = f(x + ph).$$

$$f(x_0 + ph) = E^p f(x_0) = (1 + \Delta)^p f(x_0)$$

$$= \left[1 + p\Delta + \frac{p(p-1)}{2!} \Delta^2 + \frac{p(p-1)(p-2)}{3!} \Delta^3 + \dots \right] f(x_0)$$

$$\begin{aligned} f(x_0 + ph) &= f(x_0) + p\Delta f(x_0) \\ &+ \frac{p(p-1)}{2!} \Delta^2 f(x_0) + \frac{p(p-1)(p-2)}{3!} \Delta^3 f(x_0) \\ &+ \dots + \frac{p(p-1)\cdots(p-n+1)}{n!} \Delta^n f(x_0) + \text{Error} \end{aligned}$$

This is known as Newton's forward difference formula for interpolation, which gives the value of $f(x_0 + ph)$ in terms of $f(x_0)$ and its leading differences.

This formula is also known as Newton-Gregory forward difference interpolation formula. Here $p=(x-x_0)/h$.

An alternate expression is

$$\begin{aligned} y_x = & y_0 + p\Delta y_0 + \frac{p(p-1)}{2!} \Delta^2 y_0 \\ & + \frac{p(p-1)(p-2)}{3!} \Delta^3 y_0 + \dots \\ & + \frac{p(p-1)(p-n+1)}{n!} \Delta^n y_0 + \text{Error} \end{aligned}$$

If we retain $(r + 1)$ terms, we obtain a polynomial of degree r agreeing with y_x at $x_0, x_1, \dots, x_r$.

This formula is mainly used for interpolating the values of y near the beginning of a set of tabular values and for extrapolating values of y , a short distance backward from

Numerical

Analysis

Lecture 21

Numerical Analysis Lecture 22

Chapter 5

Interpolation

Finite Difference Operators
Newton's Forward Difference
Interpolation Formula
Newton's Backward Difference
Interpolation Formula
Lagrange's Interpolation
Formula
Divided Differences
Interpolation in Two Dimensions
Cubic Spline Interpolation

$$\Delta^r y_i = \Delta^{r-1} y_{i+1} - \Delta^{r-1} y_i$$

$$\nabla^k y_i = \nabla^{k-1} y_i - \nabla^{k-1} y_{i-1},$$

$$i = n, (n-1), \dots, k$$

$$\delta^n y_i = \delta^{n-1} y_{i+(1/2)} - \delta^{n-1} y_{i-(1/2)}$$

Thus

$$\Delta y_x = y_{x+h} - y_x = f(x+h) - f(x)$$

$$\Delta^2 y_x = \Delta y_{x+h} - \Delta y_x$$

Similarly

$$\nabla y_x = y_x - y_{x-h} = f(x) - f(x-h)$$

$$\delta y_x = y_{x+(h/2)} - y_{x-(h/2)} = f\left(x + \frac{h}{2}\right) - f\left(x - \frac{h}{2}\right)$$

Shift operator, E

$$E f(x) = f(x + h)$$

$$E^n f(x) = f(x + nh)$$

$$E^n y_x = y_{x+nh}$$

**The inverse operator E^{-1}
is defined as**

$$E^{-1} f(x) = f(x - h)$$

Similarly,

$$E^{-n} f(x) = f(x - nh)$$

Average Operator, μ

$$\begin{aligned}\mu f(x) &= \frac{1}{2} \left[f\left(x + \frac{h}{2}\right) + f\left(x - \frac{h}{2}\right) \right] \\ &= \frac{1}{2} \left[y_{x+(h/2)} + y_{x-(h/2)} \right]\end{aligned}$$

Differential Operator, D

$$Df(x) = \frac{d}{dx} f(x) = f'(x)$$

$$D^2 f(x) = \frac{d^2}{dx^2} f(x) = f''(x)$$

Important Results

$$\Delta = E - 1$$

$$\nabla = 1 - E^{-1} = \frac{E - 1}{E}$$

$$\delta = E^{1/2} - E^{-1/2}$$

$$hD = \log E$$

$$\mu = \frac{1}{2} (E^{1/2} + E^{-1/2})$$

Newton's Forward Difference Interpolation Formula

Let $y = f(x)$ be a function which takes values $f(x_0)$, $f(x_0 + h)$, $f(x_0 + 2h)$, ..., corresponding to various equi-spaced values of x with spacing h , say x_0 , $x_0 + h$, $x_0 + 2h$, ...

Suppose, we wish to evaluate the function $f(x)$ for a value $x_0 + ph$, where p is any real number, then for any real number p , we have the operator E such that

$$E^p f(x) = f(x + ph).$$

$$f(x_0 + ph) = E^p f(x_0) = (1 + \Delta)^p f(x_0)$$

$$= \left[1 + p\Delta + \frac{p(p-1)}{2!} \Delta^2 + \frac{p(p-1)(p-2)}{3!} \Delta^3 + \dots \right] f(x_0)$$

$$\begin{aligned} f(x_0 + ph) &= f(x_0) + p\Delta f(x_0) \\ &+ \frac{p(p-1)}{2!} \Delta^2 f(x_0) + \frac{p(p-1)(p-2)}{3!} \Delta^3 f(x_0) \\ &+ \dots + \frac{p(p-1)\cdots(p-n+1)}{n!} \Delta^n f(x_0) + \text{Error} \end{aligned}$$

This is known as Newton's forward difference formula for interpolation, which gives the value of $f(x_0 + ph)$ in terms of $f(x_0)$ and its leading differences.

This formula is also known as Newton-Gregory forward difference interpolation formula. Here $p=(x-x_0)/h$.

An alternate expression is

$$\begin{aligned} y_x = & y_0 + p\Delta y_0 + \frac{p(p-1)}{2!} \Delta^2 y_0 \\ & + \frac{p(p-1)(p-2)}{3!} \Delta^3 y_0 + \dots \\ & + \frac{p(p-1)(p-n+1)}{n!} \Delta^n y_0 + \text{Error} \end{aligned}$$

Exercise

Find a cubic polynomial in x which takes on the values -3, 3, 11, 27, 57 and 107, when $x = 0, 1, 2, 3, 4$ and 5 respectively.

Solution

Here, the observations are given at equal intervals of unit width.

To determine the required polynomial, we first construct the difference table

Difference Table

x	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$
0	-3	6		
1	3	8	2	
2	11	16	8	6
3	27	30	14	6
4	57	30	20	6
5	107	50		

Since the 4th and higher order differences are zero, the required Newton's interpolation formula

$$f(x_0 + ph) = f(x_0) + p\Delta f(x_0) + \frac{p(p-1)}{2}\Delta^2 f(x_0) + \frac{p(p-1)(p-2)}{6}\Delta^3 f(x_0)$$

Here,

$$p = \frac{x - x_0}{h} = \frac{x - 0}{1} = x$$

$$\Delta f(x_0) = 6$$

$$\Delta^2 f(x_0) = 2$$

$$\Delta^3 f(x_0) = 6$$

Substituting these values into the formula, we have

$$f(x) = -3 + 6x + \frac{x(x-1)}{2} (2) \\ + \frac{x(x-1)(x-2)}{6} (6)$$

$$f(x) = x^3 - 2x^2 + 7x - 3,$$

The required cubic polynomial.

NEWTON'S BACKWARD DIFFERENCE INTERPOLATION FORMULA

For interpolating the value of the function $y = f(x)$ near the end of table of values, and to extrapolate value of the function a short distance forward from y_n , Newton's backward interpolation formula is used

Derivation

Let $y = f(x)$ be a function which takes on values $f(x_n), f(x_n-h), f(x_n-2h), \dots, f(x_0)$ corresponding to equispaced values $x_n, x_n-h, x_n-2h, \dots, x_0$. Suppose, we wish to evaluate the function $f(x)$ at $(x_n + ph)$,

**where p is any real number,
then we have the shift
operator E , such that**

$$f(x_n + ph) = E^p f(x_n) = (E^{-1})^{-p} f(x_n) = (1 - \nabla)^{-p} f(x_n)$$

Binomial expansion yields,

$$f(x_n + ph) = \left[1 + p\nabla + \frac{p(p+1)}{2!} \nabla^2 + \frac{p(p+1)(p+2)}{3!} \nabla^3 + \dots \right. \\ \left. + \frac{p(p+1)(p+2) \cdots (p+n-1)}{n!} \nabla^n + \text{Error} \right] f(x_n)$$

That is,

$$\begin{aligned} f(x_n + ph) &= f(x_n) + p\nabla f(x_n) \\ &+ \frac{p(p+1)}{2!} \nabla^2 f(x_n) \\ &+ \frac{p(p+1)(p+2)}{3!} \nabla^3 f(x_n) + \cdots \\ &+ \frac{p(p+1)(p+2)\cdots(p+n-1)}{n!} \nabla^n f(x_n) + \text{Error} \end{aligned}$$

This formula is known as Newton's backward interpolation formula. This formula is also known as Newton's-Gregory backward difference interpolation formula.

If we retain $(r + 1)$ terms, we obtain a polynomial of degree r agreeing with $f(x)$ at $x_n, x_{n-1}, \dots, x_{n-r}$. Alternatively, this formula can also be written as

$$\begin{aligned} y_x = & y_n + p \nabla y_n + \frac{p(p+1)}{2!} \nabla^2 y_n \\ & + \frac{p(p+1)(p+2)}{3!} \nabla^3 y_n + \dots \\ & + \frac{p(p+1)(p+2) \cdots (p+n-1)}{n!} \nabla^n y_n + \text{Error} \end{aligned}$$

Here

$$p = \frac{x - x_n}{h}$$

Example

For the following table of values, estimate $f(7.5)$.

x	1	2	3	4	5	6	7	8
$y = f(x)$	1	8	27	64	125	216	343	512

Solution

The value to be interpolated is at the end of the table. Hence, it is appropriate to use Newton's backward interpolation formula. Let us first construct the backward difference table for the given data

Difference Table

x	$y = f(x)$	∇y	$\nabla^2 y$	$\nabla^3 y$	$\nabla^4 y$
1	1				
2	8	7			
3	27	19	12		
4	64	37	18	6	
5	125	61	24	6	0
6	216	91	30	6	0
7	343	127	36	6	0
8	512	169	42	6	0

Since the 4th and higher order differences are zero, the required Newton's backward interpolation formula is

$$y_x = y_n + p\nabla y_n + \frac{p(p+1)}{2!} \nabla^2 y_n + \frac{p(p+1)(p+2)}{3!} \nabla^3 y_n$$

In this problem,

$$p = \frac{x - x_n}{h} = \frac{7.5 - 8.0}{1} = -0.5$$

$$\nabla y_n = 169, \quad \nabla^2 y_n = 42, \quad \nabla^3 y_n = 6$$

$$\begin{aligned} y_{7.5} &= 512 + (-0.5)(169) + \frac{(-0.5)(0.5)}{2}(42) \\ &\quad + \frac{(-0.5)(0.5)(1.5)}{6}(6) \\ &= 512 - 84.5 - 5.25 - 0.375 \\ &= 421.875 \end{aligned}$$

Example

The sales for the last five years is given in the table below. Estimate the sales for the year 1979

Year	1974	1976	1978	1980	1982
Sales (in lakhs)	40	43	48	52	57

Solution

Newton's backward difference table for the given data as

x	y	∇y	$\nabla^2 y$	$\nabla^3 y$	$\nabla^4 y$
1974	40				
1976	43	3			
1978	48	5	2		
1980	52	4	-1	-3	
1982	57	5	1	2	5

In this example,

$$p = \frac{1979 - 1982}{2} = -1.5$$

and

$$\nabla y_n = 5, \quad \nabla^2 y_n = 1,$$

$$\nabla^3 y_n = 2, \quad \nabla^4 y_n = 5$$

Newton's interpolation formula gives

$$\begin{aligned} y_{1979} = & 57 + (-1.5)5 + \frac{(-1.5)(-0.5)}{2} (1) \\ & + \frac{(-1.5)(-0.5)(0.5)}{6} (2) \\ & + \frac{(-1.5)(-0.5)(0.5)(1.5)}{24} (5) \end{aligned}$$

$$= 57 - 7.5 + 0.375 + 0.125 + 0.1172$$

Therefore,

$$y_{1979} = 50.1172$$

LAGRANGE'S INTERPOLATION FORMULA

Newton's interpolation formulae developed earlier can be used only when the values of the independent variable x are equally spaced. Also the differences of y must ultimately become small.

If the values of the independent variable are not given at equidistant intervals, then we have the basic formula associated with the name of Lagrange which will be derived now.

Numerical Analysis Lecture 22

Let $y = f(x)$ be a function which takes the values, $y_0, y_1, \dots, y_n$ corresponding to $x_0, x_1, \dots, x_n$. Since there are $(n + 1)$ values of y corresponding to $(n + 1)$ values of x , we can represent the function $f(x)$ by a polynomial of degree n .

Suppose we write this polynomial in the form .

$$f(x) = A_0x^n + A_1x^{n-1} + \cdots + A_n$$

or in the form

$$\begin{aligned} y = f(x) = & a_0(x - x_1)(x - x_2) \cdots (x - x_n) \\ & + a_1(x - x_0)(x - x_2) \cdots (x - x_n) \\ & + a_2(x - x_0)(x - x_1) \cdots (x - x_n) + \cdots \\ & + a_n(x - x_0)(x - x_1) \cdots (x - x_{n-1}) \end{aligned}$$

Here, the coefficients a_k are so chosen as to satisfy this equation by the $(n + 1)$ pairs (x_i, y_i) . Thus we get

$$y_0 = f(x_0) = a_0(x_0 - x_1)(x_0 - x_1)(x_0 - x_2) \cdots (x_0 - x_n)$$

Therefore,

$$a_0 = \frac{y_0}{(x_0 - x_1)(x_0 - x_2) \cdots (x_0 - x_n)}$$

Similarly, we obtain

$$a_1 = \frac{y_1}{(x_1 - x_0)(x_1 - x_2) \cdots (x_1 - x_n)}$$

$$a_i = \frac{y_i}{(x_i - x_0)(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)}$$

and

$$a_n = \frac{y_n}{(x_n - x_0)(x_n - x_1) \cdots (x_n - x_{n-1})}$$

**Substituting the values of
 $a_0, a_1, \dots, a_n$ we get**

$$y = f(x) = \frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)}y_0 + \frac{(x-x_0)(x-x_2)\cdots(x-x_n)}{(x_1-x_0)(x_1-x_2)\cdots(x_1-x_n)}y_1 + \cdots$$
$$+ \frac{(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)}y_i + \cdots$$
$$+ \frac{(x-x_0)(x-x_1)(x-x_2)\cdots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)(x_n-x_2)\cdots(x_n-x_{n-1})}y_n$$

**The Lagrange's formula for
interpolation**

This formula can be used whether the values $x_0, x_2, \dots, x_n$ are equally spaced or not. Alternatively, this can also be written in compact form as

$$y = f(x) = L_0(x)y_0 + L_1(x_1)y_1 + L_i(x_i)y_i + \dots + L_n(x_n)y_n$$

$$= \sum_{k=0}^n L_k(x)y_k$$

$$= \sum_{k=0}^n L_k(x)f(x_k)$$

Where,

$$L_i(x) = \frac{(x - x_0)(x - x_1) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_0)(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)}$$

We can easily observe that,

$$L_i(x_i) = 1 \quad \text{and} \quad L_i(x_j) = 0, i \neq j.$$

**Thus introducing *Kronecker*
delta notation**

$$L_i(x_j) = \delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}$$

Further, if we introduce the notation

$$\Pi(x) = \prod_{i=0}^n (x - x_i) = (x - x_0)(x - x_1) \cdots (x - x_n)$$

That is $\Pi(x)$ is a product of $(n + 1)$ factors. Clearly, its derivative $\Pi'(x)$ contains a sum of $(n + 1)$ terms in each of which one of the factors of $\Pi(x)$ will be absent.

We also define,

$$P_k(x) = \prod_{i \neq k} (x - x_i)$$

which is same as $\Pi(x)$ except that the factor $(x - x_k)$ is absent. Then

$$\Pi'(x) = P_0(x) + P_1(x) + \cdots + P_n(x)$$

But, when $x = x_k$, all terms in the above sum vanishes except $P_k(x_k)$

Hence,

$$\Pi'(x_k) = P_k(x_k) = (x_k - x_0) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)$$

$$\begin{aligned} L_k(x) &= \frac{P_k(x)}{P_k(x_k)} = \frac{P_k(x)}{\Pi'(x_k)} \\ &= \frac{\Pi(x)}{(x - x_k)\Pi'(x_k)} \end{aligned}$$

Finally, the Lagrange's interpolation polynomial of degree n can be written as

$$\begin{aligned} y(x) = f(x) &= \sum_{k=0}^n \frac{\Pi(x)}{(x - x_k) \Pi'(x_k)} f(x_k) \\ &= \sum_{k=0}^n L_k(x) f(x_k) = \sum_{k=0}^n L_k(x) y_k \end{aligned}$$

Example

Find Lagrange's interpolation polynomial fitting the points

$$y(1) = -3, y(3) = 0,$$

$$y(4) = 30, y(6) = 132.$$

Hence find $y(5)$.

Solution

The given data can be arranged as

x	1	3	4	6
$y = f(x)$	-3	0	30	132

Using Lagrange's interpolation formula, we have

$$\begin{aligned} y(x) = f(x) = & \frac{(x-3)(x-4)(x-6)}{(1-3)(1-4)(1-6)}(-3) \\ & + \frac{(x-1)(x-4)(x-6)}{(3-1)(3-4)(3-6)}(0) \\ & + \frac{(x-1)(x-3)(x-6)}{(4-1)(4-3)(4-6)}(30) \\ & + \frac{(x-1)(x-3)(x-4)}{(6-1)(6-3)(6-4)}(132) \end{aligned}$$

On simplification, we get

$$\begin{aligned} y(x) &= \frac{1}{10}(-5x^3 + 135x^2 - 460x + 300) \\ &= \frac{1}{2}(-x^3 + 27x^2 - 92x + 60) \end{aligned}$$

**which is required Lagrange's interpolation polynomial.
Now, $y(5) = 75$.**

Example

Given the following data,
evaluate $f(3)$ using
Lagrange's interpolating
polynomial.

x	1	2	5
$f(x)$	1	4	10

Solution

Using Lagrange's formula,

$$\begin{aligned} f(x) = & \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) \\ & + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) + \\ & \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2) \end{aligned}$$

Therefore

$$\begin{aligned} f(3) &= \frac{(3-2)(3-5)}{(1-2)(1-5)} (1) \\ &+ \frac{(3-1)(3-5)}{(2-1)(2-5)} (4) \\ &+ \frac{(3-1)(3-2)}{(5-1)(5-2)} (10) \\ &= 6.49999 = 6.5 \end{aligned}$$

Numerical Analysis Lecture 23

Chapter 5

Interpolation

Finite Difference Operators
Newton's Forward Difference
Interpolation Formula
Newton's Backward Difference
Interpolation Formula
Lagrange's Interpolation
Formula
Divided Differences
Interpolation in Two Dimensions
Cubic Spline Interpolation

$$\Delta^r y_i = \Delta^{r-1} y_{i+1} - \Delta^{r-1} y_i$$

$$\nabla^k y_i = \nabla^{k-1} y_i - \nabla^{k-1} y_{i-1},$$

$$i = n, (n-1), \dots, k$$

$$\delta^n y_i = \delta^{n-1} y_{i+(1/2)} - \delta^{n-1} y_{i-(1/2)}$$

Thus

$$\Delta y_x = y_{x+h} - y_x = f(x+h) - f(x)$$

$$\Delta^2 y_x = \Delta y_{x+h} - \Delta y_x$$

Similarly

$$\nabla y_x = y_x - y_{x-h} = f(x) - f(x-h)$$

$$\delta y_x = y_{x+(h/2)} - y_{x-(h/2)} = f\left(x + \frac{h}{2}\right) - f\left(x - \frac{h}{2}\right)$$

Shift operator, E

$$E f(x) = f(x + h)$$

$$E^n f(x) = f(x + nh)$$

$$E^n y_x = y_{x+nh}$$

**The inverse operator E^{-1}
is defined as**

$$E^{-1} f(x) = f(x - h)$$

Similarly,

$$E^{-n} f(x) = f(x - nh)$$

Average Operator, μ

$$\begin{aligned}\mu f(x) &= \frac{1}{2} \left[f\left(x + \frac{h}{2}\right) + f\left(x - \frac{h}{2}\right) \right] \\ &= \frac{1}{2} \left[y_{x+(h/2)} + y_{x-(h/2)} \right]\end{aligned}$$

Differential Operator, D

$$Df(x) = \frac{d}{dx} f(x) = f'(x)$$

$$D^2 f(x) = \frac{d^2}{dx^2} f(x) = f''(x)$$

Important Results

$$\Delta = E - 1$$

$$\nabla = 1 - E^{-1} = \frac{E - 1}{E}$$

$$\delta = E^{1/2} - E^{-1/2}$$

$$hD = \log E$$

$$\mu = \frac{1}{2}(E^{1/2} + E^{-1/2})$$

Newton's Forward Difference Interpolation Formula

$$\begin{aligned} f(x_0 + ph) &= f(x_0) + p\Delta f(x_0) \\ &+ \frac{p(p-1)}{2!} \Delta^2 f(x_0) + \frac{p(p-1)(p-2)}{3!} \Delta^3 f(x_0) \\ &+ \dots + \frac{p(p-1)\cdots(p-n+1)}{n!} \Delta^n f(x_0) + \text{Error} \end{aligned}$$

The Newton's forward difference formula for interpolation, which gives the value of $f(x_0 + ph)$ in terms of $f(x_0)$ and its leading differences.

This is also known as Newton-Gregory forward difference interpolation formula.

Here $p=(x-x_0)/h$.

An alternate expression is

$$\begin{aligned} y_x = & y_0 + p\Delta y_0 + \frac{p(p-1)}{2!} \Delta^2 y_0 \\ & + \frac{p(p-1)(p-2)}{3!} \Delta^3 y_0 + \dots \\ & + \frac{p(p-1)\cdots(p-n+1)}{n!} \Delta^n y_0 + \text{Error} \end{aligned}$$

NEWTON'S BACKWARD DIFFERENCE INTERPOLATION FORMULA

The formula,

$$\begin{aligned} f(x_n + ph) &= f(x_n) + p\nabla f(x_n) \\ &+ \frac{p(p+1)}{2!} \nabla^2 f(x_n) \\ &+ \frac{p(p+1)(p+2)}{3!} \nabla^3 f(x_n) + \dots \\ &+ \frac{p(p+1)(p+2)\cdots(p+n-1)}{n!} \nabla^n f(x_n) + \text{Error} \end{aligned}$$

Alternatively, this formula can also be written as

$$\begin{aligned} y_x = & y_n + p \nabla y_n + \frac{p(p+1)}{2!} \nabla^2 y_n \\ & + \frac{p(p+1)(p+2)}{3!} \nabla^3 y_n + \dots \\ & + \frac{p(p+1)(p+2) \cdots (p+n-1)}{n!} \nabla^n y_n + \text{Error} \end{aligned}$$

Here

$$p = \frac{x - x_n}{h}$$

LAGRANGE'S INTERPOLATION FORMULA

Newton's interpolation formulae developed earlier can be used only when the values of the independent variable x are equally spaced. Also the differences of y must ultimately become small.

If the values of the independent variable are not given at equidistant intervals, then we have the basic formula associated with the name of Lagrange which will be derived now.

Let $y = f(x)$ be a function which takes the values, $y_0, y_1, \dots, y_n$ corresponding to $x_0, x_1, \dots, x_n$. Since there are $(n + 1)$ values of y corresponding to $(n + 1)$ values of x , we can represent the function $f(x)$ by a polynomial of degree n .

Suppose we write this polynomial in the form .

$$f(x) = A_0x^n + A_1x^{n-1} + \cdots + A_n$$

or in the form

$$\begin{aligned} y = f(x) = & a_0(x - x_1)(x - x_2) \cdots (x - x_n) \\ & + a_1(x - x_0)(x - x_2) \cdots (x - x_n) \\ & + a_2(x - x_0)(x - x_1) \cdots (x - x_n) + \cdots \\ & + a_n(x - x_0)(x - x_1) \cdots (x - x_{n-1}) \end{aligned}$$

Here, the coefficients a_k are so chosen as to satisfy this equation by the $(n + 1)$ pairs (x_i, y_i) . Thus we get

$$y_0 = f(x_0) = a_0(x_0 - x_1)(x_0 - x_2) \cdots (x_0 - x_n)$$

Therefore,

$$a_0 = \frac{y_0}{(x_0 - x_1)(x_0 - x_2) \cdots (x_0 - x_n)}$$

Similarly, we obtain

$$a_1 = \frac{y_1}{(x_1 - x_0)(x_1 - x_2) \cdots (x_1 - x_n)}$$

$$a_i = \frac{y_i}{(x_i - x_0)(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)}$$

and

$$a_n = \frac{y_n}{(x_n - x_0)(x_n - x_1) \cdots (x_n - x_{n-1})}$$

**Substituting the values of
 $a_0, a_1, \dots, a_n$ we get**

$$y = f(x) = \frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)}y_0 + \frac{(x-x_0)(x-x_2)\cdots(x-x_n)}{(x_1-x_0)(x_1-x_2)\cdots(x_1-x_n)}y_1 + \cdots$$
$$+ \frac{(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)}y_i + \cdots$$
$$+ \frac{(x-x_0)(x-x_1)(x-x_2)\cdots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)(x_n-x_2)\cdots(x_n-x_{n-1})}y_n$$

**The Lagrange's formula for
interpolation**

This formula can be used whether the values $x_0, x_2, \dots, x_n$ are equally spaced or not. Alternatively, this can also be written in compact form as

$$y = f(x) = L_0(x)y_0 + L_1(x)y_1 + L_i(x)y_i + \cdots + L_n(x)y_n$$

$$= \sum_{k=0}^n L_k(x)y_k$$

$$= \sum_{k=0}^n L_k(x)f(x_k)$$

Where,

$$L_i(x) = \frac{(x - x_0)(x - x_1) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_0)(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)}$$

We can easily observe that,

$$L_i(x_i) = 1 \quad \text{and} \quad L_i(x_j) = 0, i \neq j.$$

**Thus introducing *Kronecker*
delta notation**

$$L_i(x_j) = \delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}$$

Further, if we introduce the notation

$$\Pi(x) = \prod_{i=0}^n (x - x_i) = (x - x_0)(x - x_1) \cdots (x - x_n)$$

That is, $\Pi(x)$ is a product of $(n + 1)$ factors. Clearly, its derivative $\Pi'(x)$ contains a sum of $(n + 1)$ terms in each of which one of the factors of $\Pi(x)$ will be absent.

We also define,

$$P_k(x) = \prod_{i \neq k} (x - x_i)$$

which is same as $\Pi(x)$ except that the factor $(x - x_k)$ is absent. Then

$$\Pi'(x) = P_0(x) + P_1(x) + \cdots + P_n(x)$$

But, when $x = x_k$, all terms in the above sum vanishes except $P_k(x_k)$

Hence,

$$\Pi'(x_k) = P_k(x_k) = (x_k - x_0) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)$$

$$\begin{aligned} L_k(x) &= \frac{P_k(x)}{P_k(x_k)} = \frac{P_k(x)}{\Pi'(x_k)} \\ &= \frac{\Pi(x)}{(x - x_k)\Pi'(x_k)} \end{aligned}$$

Finally, the Lagrange's interpolation polynomial of degree n can be written as

$$\begin{aligned} y(x) = f(x) &= \sum_{k=0}^n \frac{\Pi(x)}{(x - x_k) \Pi'(x_k)} f(x_k) \\ &= \sum_{k=0}^n L_k(x) f(x_k) = \sum_{k=0}^n L_k(x) y_k \end{aligned}$$

Example

Find Lagrange's interpolation polynomial fitting the points

$$y(1) = -3, y(3) = 0,$$

$$y(4) = 30, y(6) = 132.$$

Hence find $y(5)$.

Solution

The given data can be arranged as

x	1	3	4	6
$y = f(x)$	-3	0	30	132

Using Lagrange's interpolation formula, we have

$$\begin{aligned} y(x) = f(x) = & \frac{(x-3)(x-4)(x-6)}{(1-3)(1-4)(1-6)}(-3) \\ & + \frac{(x-1)(x-4)(x-6)}{(3-1)(3-4)(3-6)}(0) \\ & + \frac{(x-1)(x-3)(x-6)}{(4-1)(4-3)(4-6)}(30) \\ & + \frac{(x-1)(x-3)(x-4)}{(6-1)(6-3)(6-4)}(132) \end{aligned}$$

On simplification, we get

$$\begin{aligned} y(x) &= \frac{1}{10}(-5x^3 + 135x^2 - 460x + 300) \\ &= \frac{1}{2}(-x^3 + 27x^2 - 92x + 60) \end{aligned}$$

which is required Lagrange's interpolation polynomial.

Now, $y(5) = 75$.

Example

Given the following data,
evaluate $f(3)$ using Lagrange's
interpolating polynomial.

x	1	2	5
$f(x)$	1	4	10

Solution

Using Lagrange's formula,

$$\begin{aligned} f(x) = & \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) \\ & + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) \\ & + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2) \end{aligned}$$

Therefore

$$\begin{aligned} f(3) &= \frac{(3-2)(3-5)}{(1-2)(1-5)} (1) \\ &+ \frac{(3-1)(3-5)}{(2-1)(2-5)} (4) \\ &+ \frac{(3-1)(3-2)}{(5-1)(5-2)} (10) \\ &= 6.49999 = 6.5 \end{aligned}$$

DIVIDED DIFFERENCES

Numerical Analysis Lecture 23

Numerical Analysis Lecture 24

Chapter 5

Interpolation

Finite Difference Operators
Newton's Forward Difference
Interpolation Formula
Newton's Backward Difference
Interpolation Formula
Lagrange's Interpolation
Formula
Divided Differences
Interpolation in Two Dimensions
Cubic Spline Interpolation

Newton's Forward Difference Interpolation Formula

$$\begin{aligned} f(x_0 + ph) &= f(x_0) + p\Delta f(x_0) \\ &+ \frac{p(p-1)}{2!} \Delta^2 f(x_0) + \frac{p(p-1)(p-2)}{3!} \Delta^3 f(x_0) \\ &+ \dots + \frac{p(p-1)\cdots(p-n+1)}{n!} \Delta^n f(x_0) + \text{Error} \end{aligned}$$

The Newton's forward difference formula for interpolation, which gives the value of $f(x_0 + ph)$ in terms of $f(x_0)$ and its leading differences.

An alternate expression is

$$\begin{aligned} y_x = & y_0 + p\Delta y_0 + \frac{p(p-1)}{2!} \Delta^2 y_0 \\ & + \frac{p(p-1)(p-2)}{3!} \Delta^3 y_0 + \dots \\ & + \frac{p(p-1)\cdots(p-n+1)}{n!} \Delta^n y_0 + \text{Error} \end{aligned}$$

NEWTON'S BACKWARD DIFFERENCE INTERPOLATION FORMULA

The formula is,

$$\begin{aligned} f(x_n + ph) &= f(x_n) + p\nabla f(x_n) \\ &+ \frac{p(p+1)}{2!} \nabla^2 f(x_n) \\ &+ \frac{p(p+1)(p+2)}{3!} \nabla^3 f(x_n) + \dots \\ &+ \frac{p(p+1)(p+2)\cdots(p+n-1)}{n!} \nabla^n f(x_n) + \text{Error} \end{aligned}$$

**Alternatively, this formula
can also be written as**

$$\begin{aligned} y_x = & y_n + p \nabla y_n + \frac{p(p+1)}{2!} \nabla^2 y_n \\ & + \frac{p(p+1)(p+2)}{3!} \nabla^3 y_n + \dots \\ & + \frac{p(p+1)(p+2) \cdots (p+n-1)}{n!} \nabla^n y_n + \text{Error} \end{aligned}$$

Here

$$p = \frac{x - x_n}{h}$$

LAGRANGE'S INTERPOLATION FORMULA

Newton's interpolation formulae can be used only when the values of the independent variable x are equally spaced. Also the differences of y must ultimately become small.

If the values of the independent variable are not given at equidistant intervals, then we have the Lagrange formula

Here the polynomial is of the form

$$f(x) = A_0x^n + A_1x^{n-1} + \cdots + A_n$$

or in the form

$$\begin{aligned} y = f(x) = & a_0(x - x_1)(x - x_2) \cdots (x - x_n) \\ & + a_1(x - x_0)(x - x_2) \cdots (x - x_n) \\ & + a_2(x - x_0)(x - x_1) \cdots (x - x_n) + \cdots \\ & + a_n(x - x_0)(x - x_1) \cdots (x - x_{n-1}) \end{aligned}$$

Here, the coefficients a_k are so chosen as to satisfy this equation by the $(n + 1)$ pairs (x_i, y_i) . Thus we get

$$y_0 = f(x_0) = a_0(x_0 - x_1)(x_0 - x_2) \cdots (x_0 - x_n)$$

Therefore,

$$a_0 = \frac{y_0}{(x_0 - x_1)(x_0 - x_2) \cdots (x_0 - x_n)}$$

Similarly

$$a_1 = \frac{y_1}{(x_1 - x_0)(x_1 - x_2) \cdots (x_1 - x_n)}$$

$$a_i = \frac{y_i}{(x_i - x_0)(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)}$$

and

$$a_n = \frac{y_n}{(x_n - x_0)(x_n - x_1) \cdots (x_n - x_{n-1})}$$

**Substituting the values of
 $a_0, a_1, \dots, a_n$ we get**

$$y = f(x) = \frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)}y_0 + \frac{(x-x_0)(x-x_2)\cdots(x-x_n)}{(x_1-x_0)(x_1-x_2)\cdots(x_1-x_n)}y_1 + \cdots$$
$$+ \frac{(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)}y_i + \cdots$$
$$+ \frac{(x-x_0)(x-x_1)(x-x_2)\cdots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)(x_n-x_2)\cdots(x_n-x_{n-1})}y_n$$

**The Lagrange's formula for
interpolation**

This formula can be used whether the values $x_0, x_2, \dots, x_n$ are equally spaced or not. Alternatively, this can also be written in compact form as

$$y = f(x) = L_0(x)y_0 + L_1(x)y_1 + L_i(x)y_i + \cdots + L_n(x)y_n$$

$$= \sum_{k=0}^n L_k(x)y_k$$

$$= \sum_{k=0}^n L_k(x)f(x_k)$$

Where,

$$L_i(x) = \frac{(x - x_0)(x - x_1) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_0)(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)}$$

We can easily observe that,

$$L_i(x_i) = 1 \quad \text{and} \quad L_i(x_j) = 0, i \neq j.$$

**Thus introducing *Kronecker*
delta notation**

$$L_i(x_j) = \delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}$$

Further, if we introduce the notation

$$\Pi(x) = \prod_{i=0}^n (x - x_i) = (x - x_0)(x - x_1) \cdots (x - x_n)$$

That is $\Pi(x)$ is a product of $(n + 1)$ factors. Clearly, its derivative $\Pi'(x)$ contains a sum of $(n + 1)$ terms in each of which one of the factors of $\Pi(x)$ will be absent.

We also define,

$$P_k(x) = \prod_{i \neq k} (x - x_i)$$

which is same as $\Pi(x)$ except that the factor $(x - x_k)$ is absent. Then

$$\Pi'(x) = P_0(x) + P_1(x) + \cdots + P_n(x)$$

But, when $x = x_k$, all terms in the above sum vanishes except $P_k(x_k)$

Hence,

$$\Pi'(x_k) = P_k(x_k) = (x_k - x_0) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)$$

$$\begin{aligned} L_k(x) &= \frac{P_k(x)}{P_k(x_k)} = \frac{P_k(x)}{\Pi'(x_k)} \\ &= \frac{\Pi(x)}{(x - x_k)\Pi'(x_k)} \end{aligned}$$

Finally, the Lagrange's interpolation polynomial of degree n can be written as

$$\begin{aligned} y(x) = f(x) &= \sum_{k=0}^n \frac{\Pi(x)}{(x - x_k) \Pi'(x_k)} f(x_k) \\ &= \sum_{k=0}^n L_k(x) f(x_k) = \sum_{k=0}^n L_k(x) y_k \end{aligned}$$

DIVIDED DIFFERENCES

Let us assume that the function $y = f(x)$ is known for several values of x , (x_i, y_i) , for $i=0,1,..n$.

The divided differences of orders 0, 1, 2, ..., n are now defined recursively as:

$$y[x_0] = y(x_0) = y_0$$

**is the zero-th order
divided difference**

The first order divided difference is defined as

$$y[x_0, x_1] = \frac{y_1 - y_0}{x_1 - x_0}$$

Similarly, the higher order divided differences are defined in terms of lower order divided differences by the relations of the form

$$y[x_0, x_1, x_2] = \frac{y[x_1, x_2] - y[x_0, x_1]}{x_2 - x_0}$$

Generally

$$y[x_0, x_1, \dots, x_n] = \frac{1}{x_n - x_0} \begin{bmatrix} y[x_1, x_2, \dots, x_n] \\ -y[x_0, x_1, \dots, x_{n-1}] \end{bmatrix}$$

Standard format of the Divided Differences

x	$y(x)$	1 st order	2 nd order	3 rd order	4 th order
x_0	y_0				
x_1	y_1	$y[x_1, x_0]$			
x_2	y_2	$y[x_2, x_1]$	$y[x_0, x_1, x_2]$	$y[x_0, x_1, x_2, x_3]$	
x_3	y_3	$y[x_3, x_2]$	$y[x_1, x_2, x_3]$	$y[x_1, x_2, x_3, x_4]$	$y[x_0, x_1, x_2, x_3, x_4]$
x_4	y_4	$y[x_4, x_3]$	$y[x_2, x_3, x_4]$		

We can easily verify that the divided difference is a symmetric function of its arguments.

That is,

$$y[x_1, x_0] = y[x_0, x_1] = \frac{y_0}{x_0 - x_1} + \frac{y_1}{x_1 - x_0}$$

Now,

$$y[x_0, x_1, x_2] = \frac{y[x_1, x_2] - y[x_0, x_1]}{x_2 - x_0}$$
$$= \frac{1}{x_2 - x_0} \left(\frac{y_2 - y_1}{x_2 - x_1} - \frac{y_1 - y_0}{x_1 - x_0} \right)$$

Therefore

$$\begin{aligned} & y[x_0, x_1, x_2] \\ &= \frac{y_0}{(x_0 - x_1)(x_0 - x_2)} \\ &+ \frac{y_1}{(x_1 - x_0)(x_1 - x_2)} \\ &+ \frac{y_2}{(x_2 - x_0)(x_2 - x_1)} \end{aligned}$$

This is symmetric form, hence suggests the general result as

$$\begin{aligned} y[x_0, \dots, x_k] &= \frac{y_0}{(x_0 - x_1) \cdots (x_0 - x_k)} \\ &+ \frac{y_1}{(x_1 - x_0) \cdots (x_1 - x_k)} + \cdots \\ &+ \frac{y_k}{(x_k - x_0) \cdots (x_k - x_{k-1})} \end{aligned}$$

$$y[x_0, \dots, x_k] = \sum_{i=0}^k \frac{y_i}{\prod_{\substack{j=0 \\ j \neq i}}^k (x_i - x_j)}$$

NEWTON'S DIVIDED DIFFERENCE INTERPOLATION FORMULA

Let $y = f(x)$ be a function which takes values $y_0, y_1, \dots, y_n$ corresponding to $x = x_i, i = 0, 1, \dots, n$. We choose an interpolating polynomial, interpolating at $x = x_i, i = 0, 1, \dots, n$ in the following form

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

Here, the coefficients a_k are so chosen as to satisfy above equation by the $(n + 1)$ pairs (x_i, y_i) .

Thus, we have

$$\left. \begin{aligned} y(x_0) &= f(x_0) = y_0 = a_0 \\ y(x_1) &= f(x_1) = y_1 = a_0 + a_1(x_1 - x_0) \\ y(x_2) &= f(x_2) = y_2 = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) \\ &\dots \\ y_n &= a_0 + a_1(x_n - x_0) + a_2(x_n - x_0)(x_n - x_1) + \dots + a_n(x_n - x_0) \dots (x_n - x_{n-1}) \end{aligned} \right\}$$

**The coefficients
 $a_0, a_1, \dots, a_n$ can be
easily obtained from
the above system of
equations, as they
form a lower triangular
matrix.**

The first equation gives

$$a_0 = y(x_0) = y_0$$

The second equation gives

$$a_1 = \frac{y_1 - y_0}{x_1 - x_0} = y[x_0, x_1]$$

Third equation yields

$$a_2 = \frac{y_2 - y_0 - (x_2 - x_0)y[x_0, x_1]}{(x_2 - x_0)(x_2 - x_1)}$$

which can be rewritten as

$$a_2 = \frac{\left[y_2 - y_1 + \left(\frac{y_1 - y_0}{x_1 - x_0} \right) (x_1 - x_0) \right] - (x_2 - x_0)y[x_0, x_1]}{(x_2 - x_0)(x_2 - x_1)}$$

that is

$$a_2 = \frac{y_2 - y_1 + y[x_0, x_1](x_1 - x_2)}{(x_2 - x_0)(x_2 - x_1)} = \frac{y[x_1, x_2] - y[x_0, x_1]}{x_2 - x_0}$$

Thus, in terms of second order divided differences, we have

$$a_2 = y[x_0, x_1, x_2]$$

Similarly, we can show that

$$a_n = y[x_0, x_1, \dots, x_n]$$

Newton's divided difference interpolation formula can be written as

$$\begin{aligned} y = f(x) = & y_0 + (x - x_0)y[x_0, x_1] \\ & + (x - x_0)(x - x_1)y[x_0, x_1, x_2] \\ & + \dots + \\ & (x - x_0)(x - x_1) \dots (x - x_{n-1})y[x_0, x_1, \dots, x_n] \end{aligned}$$

Newton's divided differences can also be expressed in terms of forward, backward and central differences. They can be easily derived

Assuming equi-spaced values of abscissa, we have

$$y[x_0, x_1] = \frac{y_1 - y_0}{x_1 - x_0} = \frac{\Delta y_0}{h}$$

$$y[x_0, x_1, x_2] = \frac{y[x_1, x_2] - y[x_0, x_1]}{x_2 - x_0} = \frac{\frac{\Delta y_1}{h} - \frac{\Delta y_0}{h}}{2h} = \frac{\Delta^2 y_0}{2!h^2}$$

By induction, we can in general arrive at the result

$$y[x_0, x_1, \dots, x_n] = \frac{\Delta^n y_0}{n! h^n}$$

Similarly

$$y[x_0, x_1] = \frac{y_1 - y_0}{x_1 - x_0} = \frac{\nabla y_1}{h}$$

$$y[x_0, x_1, x_2] = \frac{y[x_1, x_2] - y[x_0, x_1]}{x_2 - x_0} = \frac{\frac{\nabla y_2}{h} - \frac{\nabla y_1}{h}}{2h} = \frac{\nabla^2 y_2}{2!h^2}$$

In general, we have

$$y[x_0, x_1, \dots, x_n] = \frac{\nabla^n y_n}{n! h^n}$$

Also, in terms of central differences, we have

$$y[x_0, x_1] = \frac{y_1 - y_0}{x_1 - x_0} = \frac{\delta y_{1/2}}{h}$$

$$y[x_0, x_1, x_2] = \frac{y[x_1, x_2] - y[x_0, x_1]}{x_2 - x_0} = \frac{\frac{\delta y_{3/2}}{h} - \frac{\delta y_{1/2}}{h}}{2h} = \frac{\delta^2 y_1}{2!h^2}$$

In general, we have the following pattern

$$\left. \begin{aligned} y[x_0, x_1, \dots, x_{2m}] &= \frac{\delta^{2m} y_m}{(2m)! h^{2m}} \\ y[x_0, x_1, \dots, x_{2m+1}] &= \frac{\delta^{2m+1} y_{m+(1/2)}}{(2m+1)! h^{2m+1}} \end{aligned} \right\}$$

Numerical Analysis Lecture 24

Numerical Analysis Lecture 25

Chapter 5

Interpolation

Finite Difference Operators
Newton's Forward Difference
Interpolation Formula
Newton's Backward Difference
Interpolation Formula
Lagrange's Interpolation
Formula
Divided Differences
Interpolation in Two Dimensions
Cubic Spline Interpolation

Newton's Forward Difference Interpolation Formula

$$\begin{aligned} f(x_0 + ph) &= f(x_0) + p\Delta f(x_0) \\ &+ \frac{p(p-1)}{2!} \Delta^2 f(x_0) + \frac{p(p-1)(p-2)}{3!} \Delta^3 f(x_0) \\ &+ \dots + \frac{p(p-1)\cdots(p-n+1)}{n!} \Delta^n f(x_0) + \text{Error} \end{aligned}$$

An alternate expression is

$$\begin{aligned} y_x = & y_0 + p\Delta y_0 + \frac{p(p-1)}{2!} \Delta^2 y_0 \\ & + \frac{p(p-1)(p-2)}{3!} \Delta^3 y_0 + \dots \\ & + \frac{p(p-1)\cdots(p-n+1)}{n!} \Delta^n y_0 + \text{Error} \end{aligned}$$

NEWTON'S BACKWARD DIFFERENCE INTERPOLATION FORMULA

The formula is,

$$\begin{aligned} f(x_n + ph) &= f(x_n) + p\nabla f(x_n) \\ &+ \frac{p(p+1)}{2!} \nabla^2 f(x_n) \\ &+ \frac{p(p+1)(p+2)}{3!} \nabla^3 f(x_n) + \dots \\ &+ \frac{p(p+1)(p+2)\cdots(p+n-1)}{n!} \nabla^n f(x_n) + \text{Error} \end{aligned}$$

Alternatively form

$$\begin{aligned} y_x = y_n &+ p \nabla y_n + \frac{p(p+1)}{2!} \nabla^2 y_n \\ &+ \frac{p(p+1)(p+2)}{3!} \nabla^3 y_n + \dots \\ &+ \frac{p(p+1)(p+2) \cdots (p+n-1)}{n!} \nabla^n y_n + \text{Error} \end{aligned}$$

Here

$$p = \frac{x - x_n}{h}$$

LAGRANGE'S INTERPOLATION FORMULA

The Lagrange Formula for Interpolation

$$\begin{aligned} y = f(x) = & \frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)} y_0 + \frac{(x-x_0)(x-x_2)\cdots(x-x_n)}{(x_1-x_0)(x_1-x_2)\cdots(x_1-x_n)} y_1 + \cdots \\ & + \frac{(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)} y_i + \cdots \\ & + \frac{(x-x_0)(x-x_1)(x-x_2)\cdots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)(x_n-x_2)\cdots(x_n-x_{n-1})} y_n \end{aligned}$$

Alternatively compact form

$$y = f(x) = L_0(x)y_0 + L_1(x)y_1 + L_i(x)y_i + \cdots + L_n(x)y_n$$

$$= \sum_{k=0}^n L_k(x) y_k$$

$$= \sum_{k=0}^n L_k(x) f(x_k)$$

$$L_i(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)}$$

$$L_i(x_j) = \delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}$$

Also

$$\begin{aligned} L_k(x) &= \frac{P_k(x)}{P_k(x_k)} = \frac{P_k(x)}{\Pi'(x_k)} \\ &= \frac{\Pi(x)}{(x - x_k)\Pi'(x_k)} \end{aligned}$$

Finally, the Lagrange's interpolation polynomial of degree n can be written as

$$\begin{aligned} y(x) = f(x) &= \sum_{k=0}^n \frac{\Pi(x)}{(x - x_k) \Pi'(x_k)} f(x_k) \\ &= \sum_{k=0}^n L_k(x) f(x_k) = \sum_{k=0}^n L_k(x) y_k \end{aligned}$$

DIVIDED DIFFERENCES

Let us assume that the function $y = f(x)$ is known for several values of x , (x_i, y_i) , for $i=0,1,..n$.

The divided differences of orders 0, 1, 2, ..., n are now defined recursively as:

The zero-th order divided difference

$$y[x_0] = y(x_0) = y_0$$

The first order divided difference is defined as

$$y[x_0, x_1] = \frac{y_1 - y_0}{x_1 - x_0}$$

Second order divided difference

$$y[x_0, x_1, x_2] = \frac{y[x_1, x_2] - y[x_0, x_1]}{x_2 - x_0}$$

Generally

$$y[x_0, x_1, \dots, x_n] = \frac{1}{x_n - x_0} \begin{bmatrix} y[x_1, x_2, \dots, x_n] \\ -y[x_0, x_1, \dots, x_{n-1}] \end{bmatrix}$$

Standard format of the Divided Differences

x	$y(x)$	1 st order	2 nd order	3 rd order	4 th order
x_0	y_0				
x_1	y_1	$y[x_1, x_0]$			
x_2	y_2	$y[x_2, x_1]$	$y[x_0, x_1, x_2]$	$y[x_0, x_1, x_2, x_3]$	
x_3	y_3	$y[x_3, x_2]$	$y[x_1, x_2, x_3]$	$y[x_1, x_2, x_3, x_4]$	$y[x_0, x_1, x_2, x_3, x_4]$
x_4	y_4	$y[x_4, x_3]$	$y[x_2, x_3, x_4]$		

NEWTON'S DIVIDED DIFFERENCE INTERPOLATION FORMULA

Let $y = f(x)$ be a function which takes values $y_0, y_1, \dots, y_n$ corresponding to $x = x_i, i = 0, 1, \dots, n$. We choose an interpolating polynomial, interpolating at $x = x_i, i = 0, 1, \dots, n$ in the following form

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

Here, the coefficients a_k are so chosen as to satisfy above equation by the $(n + 1)$ pairs (x_i, y_i) .

Thus, we have

$$\left. \begin{aligned} y(x_0) &= f(x_0) = y_0 = a_0 \\ y(x_1) &= f(x_1) = y_1 = a_0 + a_1(x_1 - x_0) \\ y(x_2) &= f(x_2) = y_2 = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) \\ &\dots \\ y_n &= a_0 + a_1(x_n - x_0) + a_2(x_n - x_0)(x_n - x_1) + \dots + a_n(x_n - x_0) \dots (x_n - x_{n-1}) \end{aligned} \right\}$$

The first equation gives

$$a_0 = y(x_0) = y_0$$

The second equation gives

$$a_1 = \frac{y_1 - y_0}{x_1 - x_0} = y[x_0, x_1]$$

Third equation yields

$$a_2 = \frac{y_2 - y_0 - (x_2 - x_0)y[x_0, x_1]}{(x_2 - x_0)(x_2 - x_1)}$$

which can be rewritten as

$$a_2 = \frac{\left[y_2 - y_1 + \left(\frac{y_1 - y_0}{x_1 - x_0} \right) (x_1 - x_0) \right] - (x_2 - x_0)y[x_0, x_1]}{(x_2 - x_0)(x_2 - x_1)}$$

that is

$$a_2 = \frac{y_2 - y_1 + y[x_0, x_1](x_1 - x_2)}{(x_2 - x_0)(x_2 - x_1)} = \frac{y[x_1, x_2] - y[x_0, x_1]}{x_2 - x_0}$$

Thus, in terms of second order divided differences, we have

$$a_2 = y[x_0, x_1, x_2]$$

Similarly, we can show that

$$a_n = y[x_0, x_1, \dots, x_n]$$

Newton's divided difference interpolation formula

$$\begin{aligned} y = f(x) = & y_0 + (x - x_0)y[x_0, x_1] \\ & + (x - x_0)(x - x_1)y[x_0, x_1, x_2] \\ & + \dots + \\ & (x - x_0)(x - x_1) \dots (x - x_{n-1})y[x_0, x_1, \dots, x_n] \end{aligned}$$

Newton's divided differences can also be expressed in terms of forward, backward and central differences.

Assuming equi-spaced values of abscissa, we have

$$y[x_0, x_1] = \frac{y_1 - y_0}{x_1 - x_0} = \frac{\Delta y_0}{h}$$

$$y[x_0, x_1, x_2] = \frac{y[x_1, x_2] - y[x_0, x_1]}{x_2 - x_0} = \frac{\frac{\Delta y_1}{h} - \frac{\Delta y_0}{h}}{2h} = \frac{\Delta^2 y_0}{2!h^2}$$

By induction, we can in general arrive at the result

$$y[x_0, x_1, \dots, x_n] = \frac{\Delta^n y_0}{n! h^n}$$

Similarly

$$y[x_0, x_1] = \frac{y_1 - y_0}{x_1 - x_0} = \frac{\nabla y_1}{h}$$

$$y[x_0, x_1, x_2] = \frac{y[x_1, x_2] - y[x_0, x_1]}{x_2 - x_0} = \frac{\frac{\nabla y_2}{h} - \frac{\nabla y_1}{h}}{2h} = \frac{\nabla^2 y_2}{2!h^2}$$

In general, we have

$$y[x_0, x_1, \dots, x_n] = \frac{\nabla^n y_n}{n! h^n}$$

Also, in terms of central differences, we have

$$y[x_0, x_1] = \frac{y_1 - y_0}{x_1 - x_0} = \frac{\delta y_{1/2}}{h}$$

$$y[x_0, x_1, x_2] = \frac{y[x_1, x_2] - y[x_0, x_1]}{x_2 - x_0} = \frac{\frac{\delta y_{3/2}}{h} - \frac{\delta y_{1/2}}{h}}{2h} = \frac{\delta^2 y_1}{2!h^2}$$

In general, we have the following pattern

$$\left. \begin{aligned} y[x_0, x_1, \dots, x_{2m}] &= \frac{\delta^{2m} y_m}{(2m)! h^{2m}} \\ y[x_0, x_1, \dots, x_{2m+1}] &= \frac{\delta^{2m+1} y_{m+(1/2)}}{(2m+1)! h^{2m+1}} \end{aligned} \right\}$$

Example

Find the interpolating polynomial by (i) Lagrange's formula and (ii) Newton's divided difference formula for the following data. Hence show that they represent the same interpolating polynomial.

X	0	1	2	4
Y	1	1	2	5

Solution

The divided difference table for the given data is constructed as follows:

X	Y	1 st divided difference	2 nd divided difference	3 rd divided difference
0	1			
1	1	0		
2	2	1	$1/2$	$-1/2$
4	5	$3/2$	$1/6$	

(i) Lagrange's interpolation formula gives

$$y = f(x) = \frac{(x-1)(x-2)(x-4)}{(-1)(-2)(-4)}(1) + \frac{(x-0)(x-2)(x-4)}{(1-0)(1-2)(1-4)}(1)$$

$$+ \frac{(x-0)(x-1)(x-4)}{(2)(2-1)(2-4)}(2) + \frac{(x-0)(x-1)(x-2)}{4(4-1)(4-2)}(5)$$

$$= -\frac{(x^3 - 7x^2 + 14x - 8)}{8} + \frac{x^3 - 6x^2 + 8x}{3} - \frac{x^3 - 5x^2 + 4x}{2}$$

$$+ \frac{5(x^3 - 3x^2 + 2x)}{24}$$

$$= -\frac{x^3}{12} + \frac{3x^2}{4} - \frac{2}{3}x + 1$$

(ii) Newton's divided difference formula gives

$$y = f(x) = 1 + (x-0)(0) + (x-0)(x-1)\left(\frac{1}{2}\right) + (x-0)(x-1)(x-2)\left(-\frac{1}{12}\right)$$

$$= -\frac{x^3}{12} + \frac{3x^2}{4} - \frac{2}{3}x + 1$$

We observe that the interpolating polynomial by both Lagrange's and Newton's divided difference formulae is one and the same.

Note!

**Newton's formula
involves less number of
arithmetic operations
than that of Lagrange's.**

Example

Using Newton's divided difference formula, find the quadratic equation for the following data.

Hence find $y(2)$.

X	0	1	4
y	2	1	4

Solution:

The divided difference table for the given data is constructed as:

x	y	1 st divided difference	2 nd divided difference
0	2		
1	1	-1	1/2
4	4	1	

Now, using Newton's divided difference formula, we have

$$\begin{aligned} y &= 2 + (x - 0)(-1) + (x - 0)(x - 1)\left(\frac{1}{2}\right) \\ &= \frac{1}{2}(x^2 - 3x + 4) \end{aligned}$$

Hence, $y(2) = 1$.

Example

A function $y = f(x)$ is given at the sample points $x = x_0, x_1$ and x_2 . Show that the Newton's divided difference interpolation formula and the corresponding Lagrange's interpolation formula are identical.

Solution

For the function $y = f(x)$, we have the data

$$(x_i, y_i), i = 0, 1, 2.$$

The interpolation polynomial using Newton's divided difference formula is given as

$$y = f(x) = y_0 + (x - x_0)y[x_0, x_1] + (x - x_0)(x - x_1)y[x_0, x_1, x_2]$$

Using the definition of divided differences, we can rewrite the equation in the form

$$y = y_0(x - x_0) \frac{(y_1 - y_0)}{(x_1 - x_0)} + (x - x_0)(x - x_1) \left[\frac{y_0}{(x_0 - x_1)(x_0 - x_2)} + \frac{y_1}{(x_1 - x_0)(x_1 - x_2)} + \frac{y_2}{(x_2 - x_0)(x_2 - x_1)} \right]$$

$$= \left[1 - \frac{(x_0 - x)}{(x_0 - x_1)} + \frac{(x - x_0)(x - x_1)}{(x_0 - x_1)(x_0 - x_2)} \right] y_0$$

$$+ \left[\frac{(x - x_0)}{(x_1 - x_0)} + \frac{(x - x_0)(x - x_1)}{(x_1 - x_0)(x_1 - x_2)} \right] y_1 + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} y_2$$

On simplification, it reduces to

$$y = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} y_0 + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} y_1 + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} y_2$$

which is the Lagrange's form of interpolation polynomial.

Hence two forms are identical.

Newton's Divided Difference Formula with Error Term

Following the basic definition of divided differences, we have for any x

$$y(x) = y_0 + (x - x_0)y[x, x_0]$$

$$y[x, x_0] = y[x_0, x_1] + (x - x_1)y[x, x_0, x_1]$$

$$y[x, x_0, x_1] = y[x_0, x_1, x_2] + (x - x_2)y[x, x_0, x_1, x_2]$$

$$y[x, x_0, \dots, x_{n-1}] = y[x_0, x_1, \dots, x_n] + (x - x_n)y[x, x_0, \dots, x_n]$$

**Multiplying the
second Equation by $(x - x_0)$,
third by $(x - x_0)(x - x_1)$
and so on,
and the last by
 $(x - x_0)(x - x_1) \dots (x - x_{n-1})$ and
adding the resulting
equations, we obtain**

$$y(x) = y_0 + (x - x_0)y[x_0, x_1] + (x - x_0)(x - x_1)y[x_0, x_1, x_2] + \cdots \\ + (x - x_0)(x - x_1) \cdots (x - x_{n-1})y[x_0, x_1, \dots, x_n] + \varepsilon(x)$$

Where

$$\varepsilon(x) = (x - x_0)(x - x_1) \cdots (x - x_n)y[x, x_0, \dots, x_n]$$

**Please note that for $x = x_0, x_1, \dots, x_n$,
the error term $\varepsilon(x)$ vanishes**

Numerical Analysis Lecture 25

Numerical Analysis Lecture 26

Chapter 5

Interpolation

Finite Difference Operators
Newton's Forward Difference
Interpolation Formula
Newton's Backward Difference
Interpolation Formula
Lagrange's Interpolation
Formula
Divided Differences
Interpolation in Two Dimensions
Cubic Spline Interpolation

DIVIDED DIFFERENCES

The zero-th order divided difference

$$y[x_0] = y(x_0) = y_0$$

The first order divided difference is defined as

$$y[x_0, x_1] = \frac{y_1 - y_0}{x_1 - x_0}$$

Second order divided difference

$$y[x_0, x_1, x_2] = \frac{y[x_1, x_2] - y[x_0, x_1]}{x_2 - x_0}$$

Generally

$$y[x_0, x_1, \dots, x_n] = \frac{1}{x_n - x_0} \begin{bmatrix} y[x_1, x_2, \dots, x_n] \\ -y[x_0, x_1, \dots, x_{n-1}] \end{bmatrix}$$

NEWTON'S DIVIDED DIFFERENCE INTERPOLATION FORMULA

Newton's divided difference interpolation formula

$$\begin{aligned} y = f(x) = & y_0 + (x - x_0)y[x_0, x_1] \\ & + (x - x_0)(x - x_1)y[x_0, x_1, x_2] \\ & + \dots + \\ & (x - x_0)(x - x_1) \dots (x - x_{n-1})y[x_0, x_1, \dots, x_n] \end{aligned}$$

Newton's divided differences can also be expressed in terms of forward, backward and central differences.

In terms of forward differences

$$y[x_0, x_1, \dots, x_n] = \frac{\Delta^n y_0}{n! h^n}$$

In terms of backward differences

$$y[x_0, x_1, \dots, x_n] = \frac{\nabla^n y_n}{n! h^n}$$

In terms of central differences

$$\left. \begin{aligned} y[x_0, x_1, \dots, x_{2m}] &= \frac{\delta^{2m} y_m}{(2m)! h^{2m}} \\ y[x_0, x_1, \dots, x_{2m+1}] &= \frac{\delta^{2m+1} y_{m+(1/2)}}{(2m+1)! h^{2m+1}} \end{aligned} \right\}$$

Newton's Divided Difference Formula with Error Term

Following the basic definition of divided differences, we have for any x

$$\left. \begin{aligned} y(x) &= y_0 + (x - x_0)y[x, x_0] \\ y[x, x_0] &= y[x_0, x_1] + (x - x_1)y[x, x_0, x_1] \\ y[x, x_0, x_1] &= y[x_0, x_1, x_2] + (x - x_2)y[x, x_0, x_1, x_2] \\ &\vdots \\ y[x, x_0, \dots, x_{n-1}] &= y[x_0, x_1, \dots, x_n] + (x - x_n)y[x, x_0, \dots, x_n] \end{aligned} \right\}$$

**Multiplying the
second Equation by $(x - x_0)$,
third by $(x - x_0)(x - x_1)$
and so on,
and the last by
 $(x - x_0)(x - x_1) \dots (x - x_{n-1})$ and
adding the resulting
equations, we obtain**

$$y(x) = y_0 + (x - x_0)y[x_0, x_1] + (x - x_0)(x - x_1)y[x_0, x_1, x_2] + \cdots \\ + (x - x_0)(x - x_1) \cdots (x - x_{n-1})y[x_0, x_1, \dots, x_n] + \varepsilon(x)$$

Where

$$\varepsilon(x) = (x - x_0)(x - x_1) \cdots (x - x_n)y[x, x_0, \dots, x_n]$$

**Please note that for $x = x_0, x_1, \dots, x_n$,
the error term $\varepsilon(x)$ vanishes**

Error Term in Interpolation Formulae

We know, if $y(x)$ is approximated by a polynomial $P_n(x)$ of degree n then the error is given by

$$\varepsilon(x) = y(x) - P_n(x),$$

where,

$$\varepsilon(x) = (x - x_0)(x - x_1) \cdots (x - x_n) y[x, x_0, \dots, x_n]$$

Alternatively it is also expressed as

$$\varepsilon(x) = \Pi(x)y[x, x_0, \dots, x_n] = K \Pi(x)$$

Now, consider a function $F(x)$, such that

$$F(x) = y(x) - P_n(x) - K \Pi(x)$$

Determine the constant K in such a way that $F(x)$ vanishes for $x = x_0, x_1, \dots, x_n$ and also for an arbitrarily chosen point $\bar{x}$, which is different from the given $(n + 1)$ points.

Let I denotes the closed interval spanned by the values $x_0, \dots, x_n, \bar{x}$.

Then $F(x)$ vanishes $(n + 2)$ times in the interval I .

By Rolle's theorem $F'(x)$ vanishes at least $(n + 1)$ times in the interval I , $F''(x)$ at least n times, and so on.

Eventually, we can show that $F^{(n+1)}(x)$ vanishes at least once in the interval I , say at $x = \xi$. Thus, we obtain

$$0 = y^{(n+1)}(\xi) - P_n^{(n+1)}(\xi) - K \Pi^{(n+1)}(\xi)$$

Since $P_n(x)$ is a polynomial of degree n , its $(n + 1)$ th derivative is zero. Also, from the definition of $\Pi(x)$

$$\Pi^{(n+1)}(x) = (n + 1)!.$$

Therefore, we get

$$K = \frac{y^{(n+1)}(\xi)}{(n+1)!}$$

Hence

$$\varepsilon(x) = y(x) - P_n(x) = \frac{y^{(n+1)}(\xi)}{(n+1)!} \Pi(x)$$

for some $\xi = \xi(x)$ in the interval I .

Further more, we observe that

$$y[x, x_0, \dots, x_n] = \frac{y^{(n+1)}(\xi)}{(n+1)!}$$

Thus the error committed in replacing $y(x)$ by either Newton's divided difference formula or by an identical Lagrange's formula is

$$\varepsilon(x) = \Pi(x)y[x, x_0, \dots, x_n] = \Pi(x) \frac{y^{(n+1)}(\xi)}{(n+1)!}$$

INTERPOLATION IN TWO DIMENSIONS

Let u be a polynomial function in two variables, say x and y , in particular quadratic in x and cubic in y , which in general can be written as

$$\begin{aligned} u = f(x, y) = & a_0 + a_1x + a_2y + a_3x^2 \\ & + a_4xy + a_5y^2 + a_6y^3 + a_7y^2x + a_8yx^2 \\ & + a_9y^3x + a_{10}y^2x^2 + a_{11}y^3x^2 \end{aligned}$$

This relation involves many terms. If we have to write a relation involving three or more variables, even low degree polynomials give rise to long expressions. If necessary, we can certainly write, but such complications can be avoided by handling each variable separately.

If we let x , a constant, say $x = c$, the equation simplifies to the form

$$u = \Big|_{x=c} = b_0 + b_1 y + b_2 y^2 + b_3 y^3$$

Now we adopt the following procedure to interpolate at a point $(1, m)$ in a table of two variables, by treating one variable a constant say $x = x_1$. The problem reduces to that of a single variable interpolation.

Any one of the methods discussed in preceding sections can then be applied to get $f(x_1, m)$. Then we repeat this procedure for various values of x say $x = x_2, x_3, \dots, x_n$ keeping y constant. Thus, we get a new table with y constant at the value $y = m$ and with x varying. We can then interpolate at $x = 1$.

Example

Tabulate the values of the function $f(x) = x^2 + y^2 - y$ for $x = 0, 1, 2, 3, 4$ and $y = 0, 1, 2, 3, 4$. Using the table of the values, compute $f(2.5, 3.5)$ by numerical double interpolation

Solution

The values of the function for the given values of the given values of x and y are given in the following table

X		y....			
	0	1	2	3	4
0	0	0	2	6	12
1	1	1	3	7	13
2	4	4	6	10	16
3	9	9	11	15	21
4	16	16	18	22	28

Cont!

Using quadratic interpolation in both x and y directions we need to consider three points in x and y directions. To start with we have to treat one variable as constant, say x . Keeping $x = 2.5$, $y = 3.5$ as near the center of the set, we choose the table of values corresponding to $x = 1, 2, 3$ and $y = 2, 3, 4$.

Cont !

The region of fit for the construction of our interpolation polynomial is shown in different color in the table

X		y....			
	0	1	2	3	4
0	0	0	2	6	12
1	1	1	3	7	13
2	4	4	6	10	16
3	9	9	11	15	21
4	16	16	18	22	28

Cont !

Thus using Newton's forward difference formula, we have

	At $x = 1$		
Y	f	Δf	$\Delta^2 f$
2	3		
3	7	4	2
4	13	6	

Cont !

Similarly

	At $x = 2$		
Y	f	Δf	$\Delta^2 f$
2	6		
3	10	4	2
4	16	6	

Cont !

Similarly

	At $x = 3$		
Y	f	Δf	$\Delta^2 f$
2	11		
3	15	4	2
4	21	6	

$$\text{with } p = \frac{y - y_0}{h} = \frac{3.5 - 2}{1} = 1.5$$

Cont !

$$\text{with } p = \frac{y - y_0}{h} = \frac{3.5 - 2}{1} = 1.5$$

$$\begin{aligned} f(1, 3.5) &= f_0 + p\Delta f_0 + \frac{p(p-1)}{2!} \Delta^2 f_0 \\ &= 3 + (1.5)(4) + \frac{(1.5)(0.5)}{2} (2) = 9.75 \end{aligned}$$

Cont !

$$f(2, 3.5) = 6 + (1.5)(4) + \frac{(1.5)(0.5)}{2}(2) = 12.75$$

$$f(3, 3.5) = 11 + (1.5)(4) + \frac{(1.5)(0.5)}{2}(2) = 17.75$$

Cont !

Therefore we arrive at the following result

	At $y=3.5$		
X	f	Δf	$\Delta^2 f$
1	9.75		
2	12.75	3	2
3	17.75	5	

Cont !

Now defining $p = \frac{2.5 - 1}{1} = 1.5$

$$f(2.5, 3.5) = 9.75 + (1.5)(3) + \frac{(1.5)(0.5)}{2}(2) = 15$$

Cont !

**From the functional relation,
we also find that**

$$f(2.5, 3.5) = (2.5)^2 + (3.5)^2 - 3.5 = 15$$

**And hence no error in
interpolation !!!**

Numerical Analysis Lecture 26

Numerical Analysis Lecture 27

Chapter 7

Numerical

Differentiation

and Integration

INTRODUCTION

**DIFFERENTIATION USING
DIFFERENCE OPERATORS**

**DIFFERENTIATION USING
INTERPOLATION**

**RICHARDSON'S
EXTRAPOLATION METHOD**

NUMERICAL INTEGRATION

**NEWTON-COTES
INTEGRATION FORMULAE
THE TRAPEZOIDAL RULE
(COMPOSITE FORM)
SIMPSON'S RULES
(COMPOSITE FORM)
ROMBERG'S INTEGRATION
DOUBLE INTEGRATION**

DIFFERENTIATION USING DIFFERENCE OPERATORS

Introduction

Consider a function of single variable $y = f(x)$. If the function is known and simple, we can easily obtain its derivative (s) or can evaluate its definite integral

However, if we do not know the function as such or the function is complicated and is given in a tabular form at a set of points $x_0, x_1, \dots, x_n$, we use only numerical methods for differentiation or integration of the given function.

We shall discuss numerical approximation to derivatives of functions of two or more variables in the lectures to follow when we shall talk about partial differential equations.

In what follows, we shall derive and illustrate various formulae for numerical differentiation of a function of a single variable based on finite difference operators and interpolation.

Subsequently, we shall develop Newton-Cotes formulae and related trapezoidal rule and Simpson's rule for numerical integration of a function.

DIFFERENTIATION USING DIFFERENCE OPERATORS

We assume that the function $y = f(x)$ is given for the values of the independent variable $x = x_0 + ph$, for $p = 0, 1, 2, \dots$ and so on.

To find the derivatives of such a tabular function, we proceed as follows

Case I:

**Using forward difference operator Δ
and combining equations**

$$E f(x) = f(x + h)$$

and

$$\Delta = E - 1$$

$$hD = \log E = \log(1 + \Delta)$$

Remember the Differential operator, D is known to represents the property

$$\left. \begin{aligned} Df(x) &= \frac{d}{dx} f(x) = f'(x) \\ D^2 f(x) &= \frac{d^2}{dx^2} f(x) = f''(x) \end{aligned} \right\}$$

**This would mean that
in terms of Δ :**

$$D = \frac{1}{h} \left(\Delta - \frac{\Delta^2}{2} + \frac{\Delta^3}{3} - \frac{\Delta^4}{4} + \frac{\Delta^5}{5} - \dots \right)$$

Therefore,

$$\begin{aligned} Df(x_0) &= f'(x_0) \\ &= \frac{1}{h} \left[\Delta f(x_0) - \frac{\Delta^2 f(x_0)}{2} + \frac{\Delta^3 f(x_0)}{3} \right. \\ &\quad \left. - \frac{\Delta^4 f(x_0)}{4} + \frac{\Delta^5 f(x_0)}{5} - \dots \right] = \frac{d}{dx} f(x_0) \end{aligned}$$

In other words,

$$Dy_0 = y'_0 \\ = \frac{1}{h} \left(\Delta y_0 - \frac{\Delta^2 y_0}{2} + \frac{\Delta^3 y_0}{3} - \frac{\Delta^4 y_0}{4} + \dots \right)$$

Also, we can easily verify

$$D^2 = \frac{1}{h^2} \left(\Delta - \frac{\Delta^2}{2} + \frac{\Delta^3}{3} - \frac{\Delta^4}{4} + \dots \right)^2$$
$$= \frac{1}{h^2} \left(\Delta^2 - \Delta^3 + \frac{11}{12} \Delta^4 - \frac{5}{6} \Delta^5 + \dots \right)$$

$$D^2 y_0 = \frac{d^2 y_0}{dx^2} = y_0''$$

$$= \frac{1}{h^2} \left(\Delta^2 y_0 - \Delta^3 y_0 + \frac{11}{12} \Delta^4 y_0 - \frac{5}{6} \Delta^5 y_0 + \cdots \right)$$

Case II:

**Using backward difference operator ,
we have ∇**

$$hD = -\log(1 - \nabla).$$

On expansion, we have

$$D = \frac{1}{h} \left(\nabla - \frac{\nabla^2}{2} + \frac{\nabla^3}{3} - \frac{\nabla^4}{4} + \dots \right)$$

we can also verify that

$$\begin{aligned} D^2 &= \frac{1}{h^2} \left(\nabla - \frac{\nabla^2}{2} + \frac{\nabla^3}{3} + \frac{\nabla^4}{4} + \dots \right)^2 \\ &= \frac{1}{h^2} \left(\nabla^2 + \nabla^3 + \frac{11}{12} \nabla^4 + \frac{5}{6} \nabla^5 + \dots \right) \end{aligned}$$

Hence,

$$\frac{d}{dx} y_n = Dy_n = y'_n$$
$$= \frac{1}{h} \left(\nabla y_n - \frac{\nabla^2 y_n}{2} + \frac{\nabla^3 y_n}{3} - \frac{\nabla^4 y_n}{4} + \dots \right)$$

and

$$y_n'' = D^2 y_n$$

$$= \frac{1}{h^2} \left(\nabla^2 y_n + \nabla^3 y_n + \frac{11}{12} \nabla^4 y_n + \frac{5}{6} \nabla^5 y_n + \dots \right)$$

The formulae for Dy_0 and D^2y_0 are useful to calculate the first and second derivatives at the beginning of the table of values in terms of forward differences; while formulae for y'_n and y''_n

are used to compute the first and second derivatives near the end points of the table, in terms of backward differences. Similar formulae can also be derived for computing higher order derivatives.

Case III: Using central difference operator δ and following the definitions of differential operator D , central difference operator δ and the shift operator E , we have

$$\begin{aligned}\delta &= E^{1/2} - E^{-1/2} = e^{hD/2} - e^{-hD/2} \\ &= 2 \sinh \frac{hD}{2}\end{aligned}$$

Therefore, we have

$$\frac{hD}{2} = \sinh^{-1} \frac{\delta}{2}$$

But,

$$\sinh^{-1} x = x - \frac{1}{2} \frac{x^3}{3} + \frac{1 \times 3}{2 \times 4} \frac{x^5}{5} - \frac{1 \times 3 \times 5}{2 \times 4 \times 6} \frac{x^7}{7} + \dots$$

**Using the last two equations
we get**

$$\frac{hD}{2} = \left(\frac{\delta}{2} - \frac{\delta^3}{6 \times 8} + \frac{3}{40 \times 32} \delta^5 - \dots \right)$$

That is,

$$D = \frac{1}{h} \left(\delta - \frac{1}{24} \delta^3 + \frac{3}{640} \delta^5 - \dots \right)$$

Therefore,

$$\frac{d}{dx} y = y' = Dy$$

$$= \frac{1}{h} \left(\delta y - \frac{1}{24} \delta^3 y + \frac{3}{640} \delta^5 y - \dots \right)$$

Also

$$D^2 = \frac{1}{h^2} \left(\delta^2 - \frac{1}{12} \delta^4 + \frac{3}{90} \delta^6 - \dots \right)$$

Hence

$$y'' = D^2 y = \frac{1}{h^2} \left(\delta^2 y - \frac{1}{12} \delta^4 y + \frac{1}{90} \delta^6 y - \dots \right)$$

For calculating the second derivative at an interior tabular point, we use the equation

$$D^2 = \frac{1}{h^2} \left(\delta^2 - \frac{1}{12} \delta^4 + \frac{3}{90} \delta^6 - \dots \right)$$

while for computing the first derivative at an interior tabular point, we in general use another convenient form for D , which is derived as follows. Multiply the right hand side of

$$\frac{d}{dx} y = y' = Dy$$

$$= \frac{1}{h} \left(\delta y - \frac{1}{24} \delta^3 y + \frac{3}{640} \delta^5 y - \dots \right)$$

by

$$\frac{\mu}{\sqrt{1 + (\delta^2/4)}}$$

**which is unity and noting the
Binomial expansion**

$$\left(1 + \frac{1}{4}\delta^2\right)^{-1/2} = 1 - \frac{1}{8}\delta^2 +$$

$$\frac{3}{128}\delta^4 - \frac{15}{48 \times 64}\delta^6 + \dots$$

We get

$$D = \frac{\mu}{h} \left(1 - \frac{1}{8} \delta^2 + \frac{3}{128} \delta^4 - \dots \right) \left(\delta - \frac{1}{24} \delta^3 + \frac{3}{640} \delta^5 - \dots \right)$$

On simplification, we obtain

$$D = \frac{\mu}{h} \left(\delta - \frac{1}{6} \delta^3 + \frac{4}{120} \delta^5 - \dots \right)$$

Therefore the equation can also be written in another useful form as

$$y' = D = \frac{\mu}{h} \left(\delta y - \frac{1}{6} \delta^3 y + \frac{1}{30} \delta^5 y - \dots \right)$$

The last two equations for y'' and y' respectively are known as Stirling's formulae for computing the derivatives of a tabular function. Similar formulae can be derived for computing higher order derivatives of a tabular function.

The equation for y' can also be written as

$$y' = \frac{\mu}{h} \left[\delta y - \frac{1^2}{3!} \delta^3 y + \frac{(1)^2 (2)^2}{5!} \delta^5 y - \frac{(1)^2 (2)^2 (3)^2}{7!} \delta^7 y + \dots \right]$$

In order to illustrate the use of formulae derived so far, for computing the derivatives of a tabulated function, we shall consider some examples in the next lecture.

Numerical Analysis Lecture 27

Numerical Analysis Lecture 28

Chapter 7

Numerical

Differentiation

and Integration

INTRODUCTION
DIFFERENTIATION USING
DIFFERENCE OPERATORS
DIFFERENTIATION USING
INTERPOLATION
RICHARDSON'S
EXTRAPOLATION METHOD
NUMERICAL INTEGRATION

NEWTON-COTES
INTEGRATION FORMULAE
THE TRAPEZOIDAL RULE
(COMPOSITE FORM)
SIMPSON'S RULES
(COMPOSITE FORM)
ROMBERG'S INTEGRATION
DOUBLE INTEGRATION

DIFFERENTIATION USING DIFFERENCE OPERATORS:

Applications:

Remember Using forward difference operator Δ , the shift operator, the backward difference operator and the average difference operator, we obtained the following formulae:

$$Dy_0 = y'_0$$

$$= \frac{1}{h} \left(\Delta y_0 - \frac{\Delta^2 y_0}{2} + \frac{\Delta^3 y_0}{3} - \frac{\Delta^4 y_0}{4} + \dots \right)$$

$$D^2 y_0 = \frac{d^2 y_0}{dx^2} = y''_0$$

$$= \frac{1}{h^2} \left(\Delta^2 y_0 - \Delta^3 y_0 + \frac{11}{12} \Delta^4 y_0 - \frac{5}{6} \Delta^5 y_0 + \dots \right)$$

$$\frac{d}{dx}y_n = Dy_n = y'_n$$

$$= \frac{1}{h} \left(\nabla y_n - \frac{\nabla^2 y_n}{2} + \frac{\nabla^3 y_n}{3} - \frac{\nabla^4 y_n}{4} + \dots \right)$$

$$y''_n = D^2 y_n$$

$$= \frac{1}{h^2} \left(\nabla^2 y_n + \nabla^3 y_n + \frac{11}{12} \nabla^4 y_n + \frac{5}{6} \nabla^5 y_n + \dots \right)$$

$$y'' = D^2 y = \frac{1}{h^2} \left(\delta^2 y - \frac{1}{12} \delta^4 y + \frac{1}{90} \delta^6 y - \dots \right)$$

$$\frac{d}{dx} y = y' = Dy$$

$$= \frac{1}{h} \left(\delta y - \frac{1}{24} \delta^3 y + \frac{3}{640} \delta^5 y - \dots \right)$$

Recall from what we mentioned in the last lecture that for calculating the second derivative at an interior tabular point, we use the equation

$$D^2 = \frac{1}{h^2} \left(\delta^2 - \frac{1}{12} \delta^4 + \frac{3}{90} \delta^6 - \dots \right)$$

while for computing the first derivative at an interior tabular point, we in general use another convenient form for D, which is derived as follows. Multiply the right hand side of

$$\frac{d}{dx} y = y' = Dy$$

$$= \frac{1}{h} \left(\delta y - \frac{1}{24} \delta^3 y + \frac{3}{640} \delta^5 y - \dots \right)$$

by

$$\frac{\mu}{\sqrt{1 + (\delta^2/4)}}$$

**which is unity and noting the
Binomial expansion**

$$\left(1 + \frac{1}{4}\delta^2\right)^{-1/2} = 1 - \frac{1}{8}\delta^2 + \frac{3}{128}\delta^4 - \frac{15}{48 \times 64}\delta^6 + \dots$$

we get

$$D = \frac{\mu}{h} \left(1 - \frac{1}{8} \delta^2 + \frac{3}{128} \delta^4 - \dots \right) \left(\delta - \frac{1}{24} \delta^3 + \frac{3}{640} \delta^5 - \dots \right)$$

simplification we get

$$D = \frac{\mu}{h} \left(\delta - \frac{1}{6} \delta^3 + \frac{4}{120} \delta^5 - \dots \right)$$

Therefore the equation can also be written in another useful form as

$$y' = D = \frac{\mu}{h} \left(\delta y - \frac{1}{6} \delta^3 y + \frac{1}{30} \delta^5 y - \dots \right)$$

The last two equations y'' and y' respectively are known as Stirling's formulae for computing the derivatives of a tabular function. Similar formulae can be derived for computing higher order derivatives of a tabular function.

The equation for y' can also be written as

$$y' = \frac{\mu}{h} \left[\delta y - \frac{1^2}{3!} \delta^3 y + \frac{(1)^2 (2)^2}{5!} \delta^5 y - \frac{(1)^2 (2)^2 (3)^2}{7!} \delta^7 y + \dots \right]$$

In order to illustrate the use of formulae derived so far, for computing the derivatives of a tabulated function, we consider the Following example :

Example:

Compute $f''(0)$ and $f'(0.2)$

**from the following
tabular data.**

x	$f(x)$
0.0	1.00
0.2	1.16
0.4	3.56
0.6	13.96
0.8	41.96
1.0	101.00

Solution **Since $x = 0$ and 0.2 appear at and near beginning of the table, it is appropriate to use formulae based on forward differences to find the derivatives. The difference table for the given data is depicted below:**

x	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$	$\Delta^4 f(x)$	$\Delta^5 f(x)$
0.0	1.00					
0.2	1.16	0.16				
0.4	3.56	2.40	2.24			
0.6	13.96	10.40	8.00	5.76	3.84	
0.8	41.96	28.00	17.60	9.60	3.84	0.0
1.0	101.00	59.04	31.04	13.44		

Using forward
difference formula
for $D^2 f(x)$,

i.e

$$D^2 f(x) = \frac{1}{h^2} \left[\Delta^2 f(x) - \Delta^3 f(x) + \frac{11}{12} \Delta^4 f(x) - \frac{5}{6} \Delta^5 f(x) \right]$$

We obtain

$$f''(0) = \frac{1}{(0.2)^2} \left[2.24 - 5.76 + \frac{11}{12}(3.84) - \frac{5}{6}(0) \right] = 0.0$$

Also, using the formula for $Df(x)$ we have

$$Df(x) = \frac{1}{h} \left[\Delta f(x) - \frac{\Delta^2 f(x)}{2} + \frac{\Delta^3 f(x)}{3} - \frac{\Delta^4 f(x)}{4} \right]$$

Hence,

$$f'(0.2) = \frac{1}{0.2} \left(2.40 - \frac{8.00}{2} + \frac{9.60}{3} - \frac{3.84}{4} \right) = 3.2$$

Example Find $y'(2.2)$
and $y''(2.2)$ from the
table

x	$y(x)$
1.4	4.0552
1.6	4.9530
1.8	6.0496
2.0	7.3891
2.2	9.0250

Solution:

Since $x=2.2$ occurs at the end of the table, it is appropriate to use backward difference formulae for derivatives. The backward difference table for the given data is shown below:

x	$y(x)$	∇y	$\nabla^2 y$	$\nabla^3 y$	$\nabla^4 y$

1.4	4.0552				
1.6	4.9630	0.8978			
1.8	6.0496	1.0966	0.1988		
2.0	7.3891	1.3395	0.2429	0.0441	
2.2	9.0250	1.6359	0.2964	0.0535	0.0094

Using backward difference formulae for $y'(x)$ and $y''(x)$, we have

$$y'_n = \frac{1}{h} \left(\nabla y_n + \frac{\nabla^2 y_n}{2} + \frac{\nabla^3 y_n}{3} + \frac{\nabla^4 y_n}{4} \right)$$

Therefore,

$$y'(2.2) = \frac{1}{0.2} \left(1.6359 + \frac{0.2964}{2} + \frac{0.0535}{3} + \frac{0.0094}{4} \right) = 5(1.8043) = 9.0215$$

Also

$$y_n'' = \frac{1}{h^2} \left(\nabla^2 y_n + \nabla^3 y_n + \frac{11}{12} \nabla^4 y_n \right)$$

Therefore,

$$y''(2.2) = \frac{1}{(0.2)^2} \left(0.2964 + 0.0535 + \frac{11}{12}(0.0094) \right) \\ = 25(0.3585) = 8.9629$$

Case IV: Derivation of Two and three point formulae:
Retaining only the first term in equation :

$$Dy_0 = y'_0 = \frac{1}{h} \left(\Delta y_0 - \frac{\Delta^2 y_0}{2} + \frac{\Delta^3 y_0}{3} - \frac{\Delta^4 y_0}{4} + \dots \right)$$

**we can get another useful
form for the first derivative as**

$$\begin{aligned} y'_i &= \frac{\Delta y_i}{h} = \frac{y_{i+1} - y_i}{h} \\ &= \frac{y(x_i + h) - y(x_i)}{h} \end{aligned}$$

Similarly, by retaining only the first term in Eqn.

$$\frac{d}{dx} y_n = Dy_n = y'_n = \frac{1}{h} \left(\nabla y_n - \frac{\nabla^2 y_n}{2} + \frac{\nabla^3 y_n}{3} - \frac{\nabla^4 y_n}{4} + \dots \right)$$

$$y'_i = \frac{\nabla y_i}{h} = \frac{y_i - y_{i-1}}{h}$$

$$= \frac{y(x_i) - y(x_i - h)}{h}$$

Adding the last two equations, we have

$$y'_i = \frac{y(x_i + h) - y(x_i - h)}{2h}$$

These Equations constitute two-point formulae for the first derivative. By retaining only the first term in Equation ,

$$D^2 y_0 = \frac{d^2 y_0}{dx^2} = y_0'' = \frac{1}{h^2} \left(\Delta^2 y_0 - \Delta^3 y_0 + \frac{11}{12} \Delta^4 y_0 \right. \\ \left. + \frac{5}{6} \Delta^5 y_0 + \dots \right)$$

We get

$$\begin{aligned} y_i'' &= \frac{\Delta^2 y_i}{h^2} = \frac{y_{i+2} - 2y_{i+1} + y_i}{h^2} \\ &= \frac{y(x_i + 2h) - 2y(x_i + h) + y(x_i)}{h^2} \end{aligned}$$

Similarly, we have

$$\begin{aligned} y_i'' &= \frac{\Delta^2 y_i}{h^2} \\ &= \frac{y(x_i) - 2y(x_i - h) + y(x_i - 2h)}{h^2} \end{aligned}$$

While retaining only the first term in the expression for y'' in terms of δ we obtain

$$\begin{aligned} y_i'' &= \frac{\delta^2 y_i}{h^2} = \frac{\delta y_{i+(1/2)} - \delta y_{i-(1/2)}}{h^2} \\ &= \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} \\ &= \frac{y(x_i - h) - 2y(x_i) + y(x_i + h)}{h^2} \end{aligned}$$

The last three equations constitute three-point formulae for computing the second derivative. We shall see later that these two- and three-point formulae become handy for developing extrapolation methods to numerical differentiation and integration.

Numerical Analysis Lecture 28

Numerical Analysis Lecture 29

Chapter 7

Numerical Differentiation and Integration

INTRODUCTION

DIFFERENTIATION USING

DIFFERENCE OPERATORS

DIFFERENTIATION USING

INTERPOLATION

RICHARDSON'S

EXTRAPOLATION METHOD

NUMERICAL INTEGRATION

NEWTON-COTES
INTEGRATION FORMULAE
THE TRAPEZOIDAL RULE
(COMPOSITE FORM)
SIMPSON'S RULES
(COMPOSITE FORM)
ROMBERG'S INTEGRATION
DOUBLE INTEGRATION

DIFFERENTIATION USING DIFFERENCE OPERATORS:

Applications:

Remember Using forward difference operator Δ , the shift operator, the backward difference operator and the average difference operator, we obtained several formulae and we worked out the following example :

Example:

Compute $f''(0)$ and $f'(0.2)$
from the following tabular data.

x $f(x)$ **0.0****1.00****0.2****1.16****0.4****3.56****0.6****13.96****0.8****41.96****1.0****101.00**

Solution **Since $x = 0$ and 0.2 appear at and near beginning of the table, it is appropriate to use formulae based on forward differences to find the derivatives. The difference table for the given data is depicted below:**

x	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$	$\Delta^4 f(x)$	$\Delta^5 f(x)$
0.0	1.00					
0.2	1.16	0.16				
0.4	3.56	2.40	2.24			
0.6	13.96	10.40	8.00	5.76		
0.8	41.96	28.00	17.60	9.60	3.84	
1.0	101.00	59.04	31.04	13.44	3.84	0.0

Example Find $y'(2.2)$ and $y''(2.2)$
from the table

x	y(x)
1.4	4.0552
1.6	4.9530
1.8	6.0496
2.0	7.3891
2.2	9.0250

Solution:

Since $x=2.2$ occurs at the end of the table, it is appropriate to use backward difference formulae for derivatives. The backward difference table for the given data is shown below:

x	$y(x)$	∇y	$\nabla^2 y$	$\nabla^3 y$	$\nabla^4 y$

1.4	4.0552				

1.6	4.9630	0.8978			

1.8	6.0496	1.0966	0.1988		

2.0	7.3891	1.3395	0.2429	0.0441	0.0094

2.2	9.0250	1.6359	0.2964	0.0535	

2.2	9.0250				

Using backward difference formulae for $y'(x)$ and $y''(x)$, we have

$$y'_n = \frac{1}{h} \left(\nabla y_n + \frac{\nabla^2 y_n}{2} + \frac{\nabla^3 y_n}{3} + \frac{\nabla^4 y_n}{4} \right)$$

Therefore,

$$y'(2.2) = \frac{1}{0.2} \left(1.6359 + \frac{0.2964}{2} + \frac{0.0535}{3} + \frac{0.0094}{4} \right) = 5(1.8043) = 9.0215$$

Also

$$y''_n = \frac{1}{h^2} \left(\nabla^2 y_n + \nabla^3 y_n + \frac{11}{12} \nabla^4 y_n \right)$$

Therefore,

$$y''(2.2) = \frac{1}{(0.2)^2} \left(0.2964 + 0.0535 + \frac{11}{12}(0.0094) \right) \\ = 25(0.3585) = 8.9629$$

Case IV: Derivation of two and three point formulae:

Retaining only the first term in equation :

$$Dy_0 = y'_0 = \frac{1}{h} \left(\Delta y_0 - \frac{\Delta^2 y_0}{2} + \frac{\Delta^3 y_0}{3} - \frac{\Delta^4 y_0}{4} + \dots \right)$$

**we can get another useful
form for the first derivative as**

$$\begin{aligned} y'_i &= \frac{\Delta y_i}{h} = \frac{y_{i+1} - y_i}{h} \\ &= \frac{y(x_i + h) - y(x_i)}{h} \end{aligned}$$

Similarly, by retaining only the first term in Equation.

$$\frac{d}{dx} y_n = Dy_n = y'_n = \frac{1}{h} \left(\nabla y_n - \frac{\nabla^2 y_n}{2} + \frac{\nabla^3 y_n}{3} - \frac{\nabla^4 y_n}{4} + \dots \right)$$

$$y'_i = \frac{\nabla y_i}{h} = \frac{y_i - y_{i-1}}{h}$$

$$= \frac{y(x_i) - y(x_i - h)}{h}$$

**Adding the last two equations,
we have**

$$y'_i = \frac{y(x_i + h) - y(x_i - h)}{2h}$$

These Equations constitute two-point formulae for the first derivative. By retaining only the first term in Equation ,

$$D^2 y_0 = \frac{d^2 y_0}{dx^2} = y_0'' = \frac{1}{h^2} \left(\Delta^2 y_0 - \Delta^3 y_0 + \frac{11}{12} \Delta^4 y_0 \right. \\ \left. + \frac{5}{6} \Delta^5 y_0 + \dots \right)$$

We get

$$\begin{aligned} y_i'' &= \frac{\Delta^2 y_i}{h^2} = \frac{y_{i+2} - 2y_{i+1} + y_i}{h^2} \\ &= \frac{y(x_i + 2h) - 2y(x_i + h) + y(x_i)}{h^2} \end{aligned}$$

Similarly, we have

$$\begin{aligned} y_i'' &= \frac{\Delta^2 y_i}{h^2} \\ &= \frac{y(x_i) - 2y(x_i - h) + y(x_i - 2h)}{h^2} \end{aligned}$$

While retaining only the first term in the expression for y'' in terms of δ we obtain

$$\begin{aligned}y_i'' &= \frac{\delta^2 y_i}{h^2} = \frac{\delta y_{i+(1/2)} - \delta y_{i-(1/2)}}{h^2} \\&= \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} \\&= \frac{y(x_i - h) - 2y(x_i) + y(x_i + h)}{h^2}\end{aligned}$$

The last three equations constitute three-point formulae for computing the second derivative. We shall see later that these two- and three-point formulae become handy for developing extrapolation methods to numerical differentiation and integration.

Example : Write first derivative of $f(x)$ at $x = 0.1, 0.2, 0.3$ where $f(x)$ is given by

x	0.1	0.2	0.3	0.4	0.5	0.6
$f(x)$	0.425	0.475	0.400	0.450	0.525	0.575

Using the two point equation we have

$$f'(0.1) = \frac{f(0.2) - f(0.1)}{h} = \frac{0.475 - 0.425}{0.1} = \frac{0.050}{0.1} = 0.5.$$

$$f'(0.2) = \frac{f(0.3) - f(0.2)}{h} = \frac{0.400 - 0.475}{0.1} = \frac{-0.075}{0.1} = -0.75.$$

$$f'(0.3) = \frac{f(0.4) - f(0.3)}{h} = \frac{0.450 - 0.400}{0.1} = \frac{0.05}{0.1} = 0.5.$$

Example 2: Find the 2nd derivative $f''(x)$ at 0.3, 0.4, 0.5 for the function given in the example above.

$$f''(0.3) = \frac{f(0.4) - 2f(0.3) + f(0.2)}{h^2 = 0.01} = \frac{0.125}{0.01} = 12.5$$

$$f''(0.4) = \frac{f(0.5) - 2f(0.4) + f(0.3)}{h^2 = 0.01} = \frac{0.025}{0.01} = 2.5$$

$$f''(0.5) = \frac{f(0.6) - 2f(0.5) + f(0.4)}{0.01} = \frac{0.075}{0.01} = 7.5$$

DIFFERENTIATION USING INTERPOLATION

If the given tabular function $y(x)$ is reasonably well approximated by a polynomial $P_n(x)$ of degree n , it is hoped that the result of $P'_n(x)$ will also satisfactorily approximate the corresponding derivative of $y(x)$.

However, even if $P_n(x)$ and $y(x)$ coincide at the tabular points, their derivatives or slopes may substantially differ at these points as is illustrated in the Figure below:

Y

$P_n(x)$

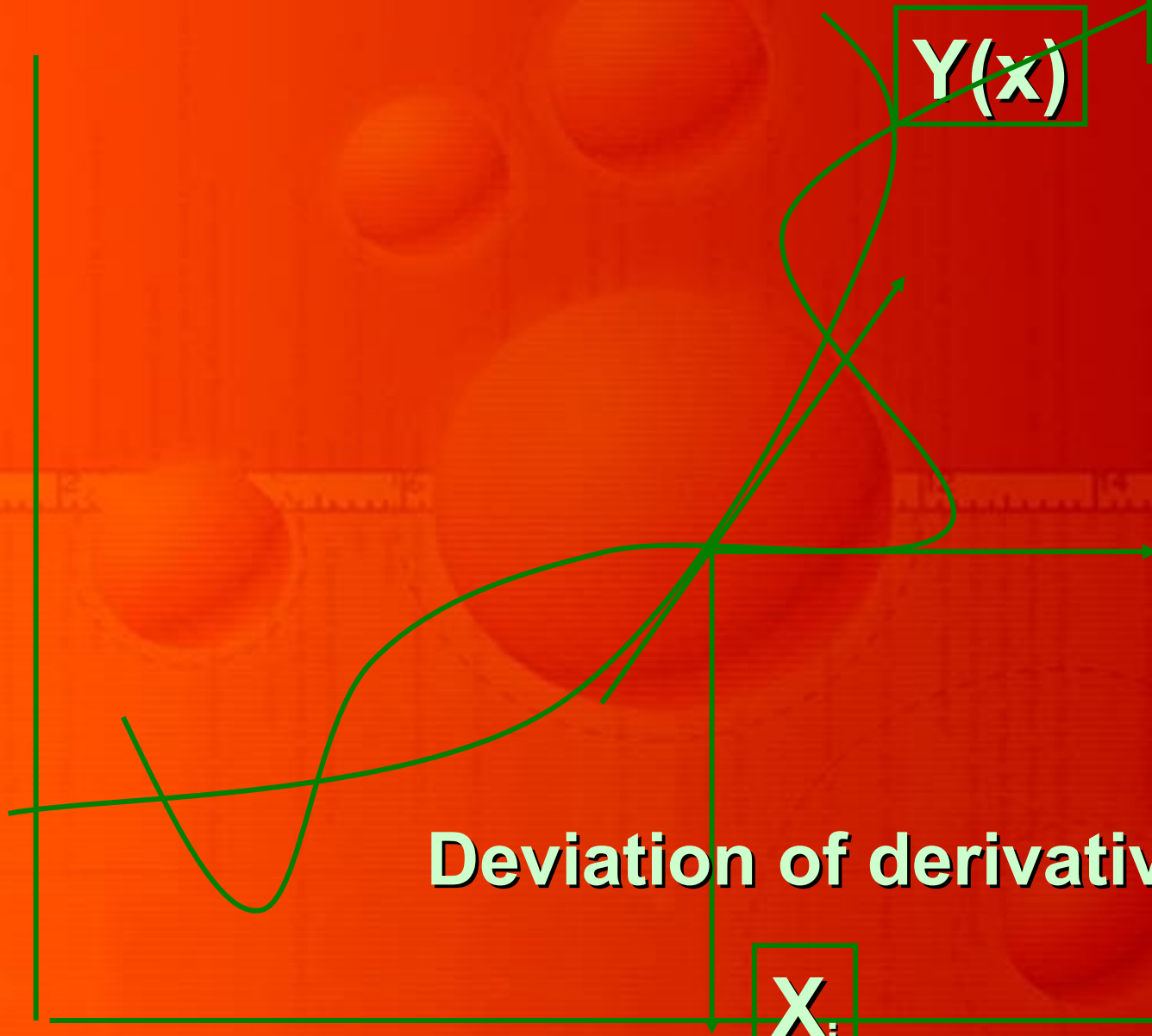
$Y(x)$

Deviation of derivatives

O

x_i

X



For higher order derivatives, the deviations may be still worst. However, we can estimate the error involved in such an approximation.

For non-equidistant tabular pairs (x_i, y_i) , $i = 0, \dots, n$ we can fit the data by using either Lagrange's interpolating polynomial or by using Newton's divided difference interpolating polynomial. In view of economy of computation, we prefer the use of the latter polynomial.

**Thus, recalling the
Newton's divided
difference interpolating
polynomial for fitting this
data as**

$$\begin{aligned} P_n(x) = & y[x_0] + (x - x_0)y[x_0, x_1] \\ & + (x - x_0)(x - x_1)y[x_0, x_1, x_2] \\ & + \cdots + \prod_{i=0}^{n-1} (x - x_i)y[x_0, x_1, \dots, x_n] \end{aligned}$$

Assuming that $P_n(x)$ is a good approximation to $y(x)$, the polynomial approximation to $y'(x)$ can be obtained by differentiating $P_n(x)$. Using product rule of differentiation, the derivative of the products in $P_n(x)$ can be seen as follows:

$$\frac{d}{dx} \prod_{i=0}^{n-1} (x - x_i)$$

$$= \sum_{i=0}^{n-1} \frac{(x - x_0)(x - x_1) \cdots (x - x_n)}{x - x_i}$$

Thus, $y'(x)$ is approximated by $P'_n(x)$ which is given by

$$P'_n(x) = y[x_0, x_1] + [(x - x_1) + (x - x_0)]y[x_0, x_1, x_2] + \cdots +$$

$$\sum_{i=0}^{n-1} \frac{(x-x_0)(x-x_1)\cdots(x-x_{n-1})}{x-x_i}$$

$$y[x_0, x_1, \dots, x_n]$$

The error estimate in this approximation can be seen from the following.

We have seen that if $y(x)$ is approximated by $P_n(x)$, the error estimate is shown to be

$$E_n(x) = y(x) - P_n(x)$$

$$= \frac{\Pi(x)}{(n+1)!} y^{(n+1)}(\xi)$$

**Its derivative with respect to x
can be written as**

$$\begin{aligned} E'_n(x) &= y'(x) - P'_n(x) \\ &= \frac{\Pi'(x)}{(n+1)!} y^{(n+1)}(\xi) \\ &\quad + \frac{\Pi(x)}{(n+1)!} \frac{d}{dx} y^{(n+1)}(\xi) \end{aligned}$$

**Since $\xi(x)$ depends on x in
an unknown way the
derivative**

$$\frac{d}{dx} y^{(n+1)}(\xi)$$

cannot be evaluated.

However, for any of the tabular points $x = x_i$, $\prod(x)$ vanishes and the difficult term drops out. Thus, the error term in the last equation at the tabular point $x = x_i$ simplifies to

$$E'_n(x_i) = \text{Error}$$

$$= \Pi'(x_i) \frac{y^{(n+1)}(\xi)}{(n+1)!}$$

for some ξ in the interval I defined by the smallest and largest of $x, x_0, x_1, \dots, x_n$ and

$$\begin{aligned}\Pi'(x_i) &= (x_i - x_0) \cdots (x_i - x_n) \\ &= \prod_{\substack{j=0 \\ j \neq i}}^n (x_i - x_j)\end{aligned}$$

The error in the r -th derivative at the tabular points can indeed be expressed analogously.

To understand this method better, we consider the following example.

**Example Find $f'(0.25)$
and $f'(0.22)$ from the
following data using the method
based on divided differences:**

x	0.15	0.21	0.23

$y = f(x)$	0.1761	0.3222	0.3617

x	0.27	0.32	0.35

$y = f(x)$	0.4314	0.5051	0.5441

Solution We first construct divided difference table for the given data as shown below:

x	y	1st divided difference	2nd divided difference	3rd divided difference
0.15	0.1761	2.4350		
0.21	0.3222	1.9750	−5.7500	15.6250
0.23	0.3617	1.7425	−3.8750	8.1064
0.27	0.4314	1.4740	−2.9833	6.7358
0.32	0.5051	1.3000	−2.1750	
0.35	0.5441			

Using divided difference formula

$$\sum_{i=0}^{n-1} \frac{(x - x_0)(x - x_1) \cdots (x - x_{n-1})}{x - x_i}$$

$$y[x_0, x_1, \dots, x_n]$$

**from a quadratic polynomial,
we have**

$$\begin{aligned} y'(x) &= P_3'(x) \\ &= y[x_0, x_1] + \{(x - x_1) \\ &\quad + (x - x_0)\} y[x_0, x_1, x_2] \end{aligned}$$

$$+ \{(x - x_1)(x - x_2) + (x - x_0)(x - x_2) \\ + (x - x_0)(x - x_1)\} y[x_0, x_1, x_2, x_3]$$

Thus, using first, second and third differences from the table, the above equation yields

$$y'(0.25)$$

$$= 2.4350 + [(0.25 - 0.21) \\ + (0.25 - 0.15)](-5.75)$$

$$\begin{aligned} &+[(0.25 - 0.21)(0.25 - 0.23) \\ &+(0.25 - 0.15)(0.25 - 0.23) \\ &+(0.25 - 0.15)(0.25 - 0.21)] \\ &(15.625) \end{aligned}$$

Therefore,

$$\begin{aligned} f'(0.25) &= 2.4350 - 0.805 + 0.10625 \\ &= 1.7363 \end{aligned}$$

Similarly, we can show that

$$f'(0.22) = 1.9734$$

Numerical Analysis

Lecture 29

Numerical Analysis Lecture 30

Chapter 7

Numerical Differentiation and Integration

INTRODUCTION

DIFFERENTIATION USING

DIFFERENCE OPERATORS

DIFFERENTIATION USING

INTERPOLATION

RICHARDSON'S

EXTRAPOLATION METHOD

NUMERICAL INTEGRATION

NEWTON-COTES
INTEGRATION FORMULAE
THE TRAPEZOIDAL RULE
(COMPOSITE FORM)
SIMPSON'S RULES
(COMPOSITE FORM)
ROMBERG'S INTEGRATION
DOUBLE INTEGRATION

DIFFERENTIATION USING INTERPOLATION

If the given tabular function $y(x)$ is reasonably well approximated by a polynomial $P_n(x)$ of degree n , it is hoped that the result of $P'_n(x)$ will also satisfactorily approximate the corresponding derivative of $y(x)$.

However, even if $P_n(x)$ and $y(x)$ coincide at the tabular points, their derivatives or slopes may substantially differ at these points as is illustrated in the Figure below:

Y

$P_n(x)$

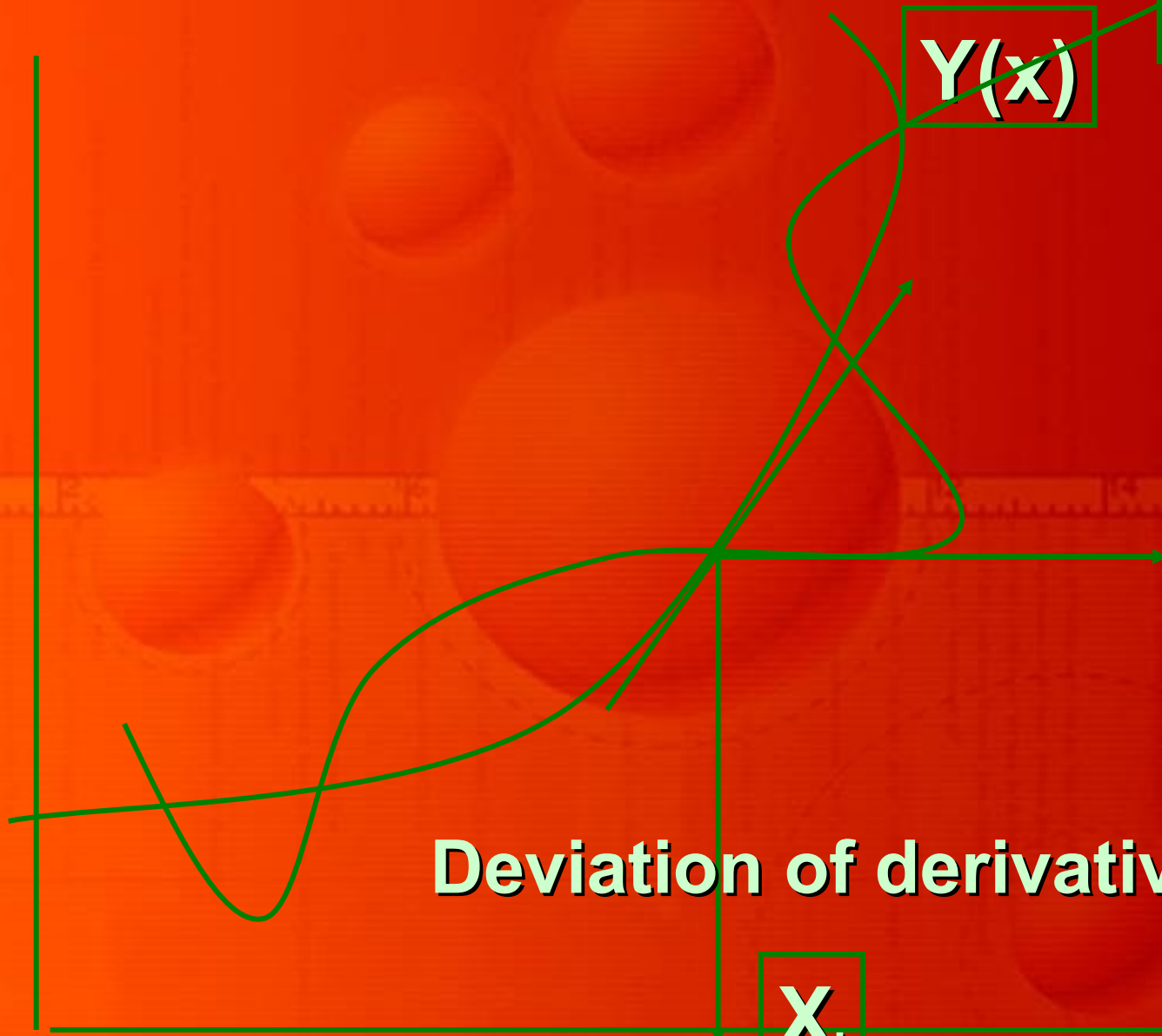
$Y(x)$

Deviation of derivatives

O

x_i

X



For higher order derivatives, the deviations may be even worst. However, we can estimate the error involved in such an approximation.

For non-equidistant tabular pairs (x_i, y_i) , $i = 0, \dots, n$ we can fit the data by using either Lagrange's interpolating polynomial or by using Newton's divided difference interpolating polynomial. In view of economy of computation, we prefer the use of the latter polynomial.

**Thus, recalling the
Newton's divided
difference interpolating
polynomial for fitting this
data as**

$$\begin{aligned} P_n(x) = & y[x_0] + (x - x_0)y[x_0, x_1] \\ & + (x - x_0)(x - x_1)y[x_0, x_1, x_2] \\ & + \cdots + \prod_{i=0}^{n-1} (x - x_i)y[x_0, x_1, \dots, x_n] \end{aligned}$$

Assuming that $P_n(x)$ is a good approximation to $y(x)$, the polynomial approximation to $y'(x)$ can be obtained by differentiating $P_n(x)$. Using product rule of differentiation, the derivative of the products in $P_n(x)$ can be seen as follows:

$$\frac{d}{dx} \prod_{i=0}^{n-1} (x - x_i)$$

$$= \sum_{i=0}^{n-1} \frac{(x - x_0)(x - x_1) \cdots (x - x_n)}{x - x_i}$$

Thus, $y'(x)$ is approximated by $P'_n(x)$ which is given by

$$P'_n(x) = y[x_0, x_1] + [(x - x_1) + (x - x_0)]y[x_0, x_1, x_2] + \cdots +$$

$$\sum_{i=0}^{n-1} \frac{(x-x_0)(x-x_1)\cdots(x-x_{n-1})}{x-x_i}$$

$$y[x_0, x_1, \dots, x_n]$$

The error estimate in this approximation can be seen from the following.

We have seen that if $y(x)$ is approximated by $P_n(x)$, the error estimate is shown to be

$$E_n(x) = y(x) - P_n(x)$$

$$= \frac{\Pi(x)}{(n+1)!} y^{(n+1)}(\xi)$$

**Its derivative with respect to x
can be written as**

$$\begin{aligned} E'_n(x) &= y'(x) - P'_n(x) \\ &= \frac{\Pi'(x)}{(n+1)!} y^{(n+1)}(\xi) \\ &\quad + \frac{\Pi(x)}{(n+1)!} \frac{d}{dx} y^{(n+1)}(\xi) \end{aligned}$$

**Since $\xi(x)$ depends on x in
an unknown way the
derivative**

$$\frac{d}{dx} y^{(n+1)}(\xi)$$

cannot be evaluated.

However, for any of the tabular points $x = x_i$, $\prod(x)$ vanishes and the difficult term drops out. Thus, the error term in the last equation at the tabular point $x = x_i$ simplifies to

$$E'_n(x_i) = \text{Error}$$

$$= \Pi'(x_i) \frac{y^{(n+1)}(\xi)}{(n+1)!}$$

for some ξ in the interval I defined by the smallest and largest of $x, x_0, x_1, \dots, x_n$ and

$$\begin{aligned}\Pi'(x_i) &= (x_i - x_0) \cdots (x_i - x_n) \\ &= \prod_{\substack{j=0 \\ j \neq i}}^n (x_i - x_j)\end{aligned}$$

The error in the r -th derivative at the tabular points can indeed be expressed analogously.

To understand this method better, we consider the following example.

**Example Find $f'(0.25)$
and $f'(0.22)$ from the
following data using the method
based on divided differences:**

x	0.15	0.21	0.23

$y = f(x)$	0.1761	0.3222	0.3617

x	0.27	0.32	0.35

$y = f(x)$	0.4314	0.5051	0.5441

Solution We first construct divided difference table for the given data as shown below:

x	y	1st divided difference	2nd divided difference	3rd divided difference
0.15	0.1761	2.4350		
0.21	0.3222	1.9750	−5.7500	15.6250
0.23	0.3617	1.7425	−3.8750	8.1064
0.27	0.4314	1.4740	−2.9833	6.7358
0.32	0.5051	1.3000	−2.1750	
0.35	0.5441			

Using divided difference formula

$$\sum_{i=0}^{n-1} \frac{(x - x_0)(x - x_1) \cdots (x - x_{n-1})}{x - x_i}$$

$$y[x_0, x_1, \dots, x_n]$$

**from a quadratic polynomial,
we have**

$$\begin{aligned} y'(x) &= P_3'(x) \\ &= y[x_0, x_1] + \{(x - x_1) \\ &\quad + (x - x_0)\} y[x_0, x_1, x_2] \end{aligned}$$

$$+ \{(x - x_1)(x - x_2) + (x - x_0)(x - x_2) \\ + (x - x_0)(x - x_1)\} y[x_0, x_1, x_2, x_3]$$

Thus, using first, second and third differences from the table, the above equation yields

$$y'(0.25)$$

$$= 2.4350 + [(0.25 - 0.21) \\ + (0.25 - 0.15)](-5.75)$$

$$\begin{aligned} &+[(0.25 - 0.21)(0.25 - 0.23) \\ &+(0.25 - 0.15)(0.25 - 0.23) \\ &+(0.25 - 0.15)(0.25 - 0.21)] \\ &(15.625) \end{aligned}$$

Therefore,

$$\begin{aligned} f'(0.25) &= 2.4350 - 0.805 + 0.10625 \\ &= 1.7363 \end{aligned}$$

Similarly, we can show that

$$f'(0.22) = 1.9734$$

RICHARDSON'S EXTRAPOLATION METHOD

To improve the accuracy of the derivative of a function, which is computed by starting with an arbitrarily selected value of h , Richardson's extrapolation method is often employed in practice, in the following manner:

Suppose we use two-point formula to compute the derivative of a function, then we have

$$y'(x) = \frac{y(x+h) - y(x-h)}{2h} + E_T$$
$$= F(h) + E_T$$

where E_T is the truncation error.
Using Taylor's series expansion, we can see that

$$E_T = c_1 h^2 + c_2 h^4 + c_3 h^6 + \dots$$

The idea of Richardson's extrapolation is to combine two computed values of $y'(x)$ using the same method but with two different step sizes usually h and $h/2$ to yield a higher order method. Thus, we have

$$y'(x) = F(h) + c_1 h^2 + c_2 h^4 + \dots$$

and

$$y'(x) = F\left(\frac{h}{2}\right) + c_1 \frac{h^2}{4} + c_2 \frac{h^4}{16} + \dots$$

Here, c_i are constants, independent of h , and $F(h)$ and $F(h/2)$ represent approximate values of derivatives. Eliminating c_1 from the above pair of equations, we get

$$y'(x)$$

$$= \frac{4F\left(\frac{h}{2}\right) - F(h)}{3} + d_1 h^4 + O(h^6)$$

Now, assuming

$$F_1\left(\frac{h}{2}\right) = \frac{4F\left(\frac{h}{2}\right) - F(h)}{3}$$

**Equation for $y'(x)$ above
reduces to**

$$y'(x) = F_1\left(\frac{h}{2}\right) + d_1 h^4 + O(h^6)$$

Thus, we have obtained a fourth-order accurate differentiation formula by combining two results which are of second-order accurate. Now, repeating the above argument, we have

$$y'(x) = F_1\left(\frac{h}{2}\right) + d_1 h^4 + O(h^6)$$

$$y'(x) = F_1\left(\frac{h}{4}\right) + \frac{d_1 h^4}{16} + O(h^6)$$

Eliminating d_1 from the above pair of equations, we get a better approximation as

$$y'(x) = F_2\left(\frac{h}{4}\right) + O(h^6)$$

**which is of sixth-order
accurate, where**

$$F_2\left(\frac{h}{4}\right) = \frac{4^2 F_1\left(\frac{h}{2^2}\right) - F_1\left(\frac{h}{2}\right)}{4^2 - 1}$$

This extrapolation process can be repeated further until the required accuracy is achieved, which is called an extrapolation to the limit. Therefore the equation for F_2 above can be generalized as

$$F_m \left(\frac{h}{2^m} \right) \\ = \frac{4^m F_{m-1} \left(\frac{h}{2^m} \right) - F_{m-1} \left(\frac{h}{2^{m-1}} \right)}{4^m - 1},$$

$$m = 1, 2, 3, \dots$$

where $F_0(h) = F(h)$.

To illustrate this procedure, we consider the following example.

Example: Using the Richardson's extrapolation limit, find $y'(0.05)$ to the function $y = -1/x$, with $h = 0.0128, 0.0064, 0.0032$.

Solution To start with, we take, $h = 0.0128$, then compute $F(h)$ as

$$\begin{aligned} F(h) &= \frac{y(x+h) - y(x-h)}{2h} \\ &= \frac{1}{0.05 + 0.0128} - \frac{1}{0.05 - 0.0128} \\ &= \frac{-}{2(0.0128)} \end{aligned}$$

$$\begin{aligned} &= \frac{-15.923566 + 26.88172}{0.0256} \\ &= 428.05289 \end{aligned}$$

**Similarly, $F(h/2) = 406.66273$.
Therefore, using Eq. (7.30), we
get**

$$F_1\left(\frac{h}{2}\right) = \frac{4F\left(\frac{h}{2}\right) - F\left(\frac{h}{2}\right)}{4 - 1}$$
$$= 399.5327$$

**which is accurate to $O(h^4)$.
Halving the step size further,
we compute**

$$\begin{aligned} & F\left(\frac{h}{2^2}\right) \\ &= \frac{-\frac{1}{0.05 + 0.0032} + \frac{1}{0.05 - 0.0032}}{2(0.0032)} \\ &= 401.64515 \end{aligned}$$

and

$$F_1\left(\frac{h}{2^2}\right)$$

$$= \frac{4F\left(\frac{h}{2^2}\right) - F\left(\frac{h}{2}\right)}{4 - 1}$$

$$= 399.97263$$

Again, using Eq. , we obtain

$$\begin{aligned} & F_2\left(\frac{h}{2^2}\right) \\ &= \frac{4^2 F_1\left(\frac{h}{2^2}\right) - F_1\left(\frac{h}{2}\right)}{4^2 - 1} \\ &= 400.00195 \end{aligned}$$

The above computation can be summarized in the following table:

h	F	$F1$	$F2$
0.0128	428.0529	399.5327	
	0.0064	406.6627	
400.00195		399.9726	
0.0032	401.6452		

Thus, after two steps, it is found that

$$y'(0.05) = 400.00915 \quad \text{while}$$

the exact value is

$$\begin{aligned} y'(0.05) &= \left(\frac{1}{x^2} \right)_{x=0.05} \\ &= \frac{1}{0.0025} = 400 \end{aligned}$$

Numerical Analysis

Lecture 30

Numerical Analysis Lecture 31

Chapter 7

Numerical Differentiation and Integration

INTRODUCTION

**DIFFERENTIATION USING
DIFFERENCE OPERATORS**

**DIFFERENTIATION USING
INTERPOLATION**

**RICHARDSON'S
EXTRAPOLATION METHOD**

NUMERICAL INTEGRATION

NEWTON-COTES
INTEGRATION FORMULAE
THE TRAPEZOIDAL RULE
(COMPOSITE FORM)
SIMPSON'S RULES
(COMPOSITE FORM)
ROMBERG'S INTEGRATION
DOUBLE INTEGRATION

Basic Issues in Integration

What does an integral represent?

$$\int_a^b f(x) dx = \textit{AREA}$$

$$\int_c^d \int_a^b g(x, y) dx dy = \textit{VOLUME}$$

Basic definition of an integral:

$$\int_a^b f(x) dx =$$

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n f(x_k) \Delta x$$

sum of Height x Width

Objective:

Evaluate $I = \int_a^b f(x) dx$ without

doing calculation analytically.

When would we want to do this?

1. Integrand is too complicated to integrate analytically.

$$\int_0^2 \frac{2 + \cos(1 + \sqrt{x})}{\sqrt{1 + 0.5x}} e^{0.5x} dx$$

2. Integrand is not precisely defined by an equation, i.e., we are given a set of data $(x_i, f(x_i))$, $i=1, \dots, n$.

All methods are applicable to integrands that are functions.

Some are applicable to tabulated values.

Key concepts:

1. Integration is a summing process. Thus virtually all numerical approximations can be represented by

$$I = \int_a^b f(x)dx = \sum_{i=0}^n \Delta x f(x_i) + E_t$$

where:

Δ **x** = weights

x_i = sampling points

E_t = truncation error

2. Closed & Open forms:

Closed forms include the end points a & b in x_i .

Open forms do not.

NUMERICAL INTEGRATION

Consider the definite integral

$$I = \int_{x=a}^b f(x) dx$$

where $f(x)$ is known either explicitly or is given as a table of values corresponding to some values of x , whether equispaced or not. Integration of such functions can be carried out using numerical techniques.

Of course, we assume that the function to be integrated is smooth and Riemann integrable in the interval of integration. In the following section, we shall develop Newton-Cotes

**formulae based on
interpolation which form the
basis for trapezoidal rule and
Simpson's rule of numerical
integration.**

NEWTON-COTES INTERGRATION FORMULAE

In this method, as in the case of numerical differentiation, we shall approximate the given tabulated function, by a polynomial $P_n(x)$ and then integrate this polynomial.

Suppose, we are given the data (x_i, y_i) , $i = 0(1)n$, at equispaced points with spacing $h = x_{i+1} - x_i$, we can represent the polynomial by any standard interpolation polynomial. Suppose, we use Lagrangian approximation, then we have

$$f(x) = \sum L_k(x) y(x_k)$$

with associated error given by

$$E(x) = \frac{\Pi(x)}{(n+1)!} y^{(n+1)}(\xi)$$

where

$$L_k(x) = \frac{\Pi(x)}{(x - x_k)\Pi'(x_k)}$$

and

$$\begin{aligned}\Pi(x) = & (x - x_0)(x - x_1) \\ & \dots (x - x_n)\end{aligned}$$

Then, we obtain an equivalent integration formula to the definite integral in the form

$$\int_a^b f(x) dx \approx \sum_{k=1}^n c_k y(x_k)$$

where c_k are the weighting coefficients given by

$$c_k = \int_a^b L_k(x) dx$$

which are also called Cotes numbers. Let the equispaced nodes are defined by

$$x_0 = a,$$

$$x_n = b,$$

$$h = \frac{b - a}{n},$$

$$x_k = x_0 + kh$$

so that $x_k - x_1 = (k - 1)h$ etc.
Now, we shall change the
variable x to p such that, $x = x_0$
+ ph , then we can rewrite
equations.

$$L_k(x) = \frac{\Pi(x)}{(x - x_k)\Pi'(x_k)}$$

$$\begin{aligned}\Pi(x) = & (x - x_0)(x - x_1) \\ & \dots (x - x_n)\end{aligned}$$

as

$$\Pi(x) = h^{n+1} p(p-1) \dots (p-n)$$

and

$$L_k(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)(x_k-x_1)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)}$$
$$= \frac{(ph)(p-1)h\cdots(p-k+1)h(p-k-1)h\cdots(p-n)h}{(kh)(k-1)h\cdots(1)(h)(-1)h\cdots(k-n)h}$$

or

$$L_k(x) = (-1)^{(n-k)}$$

$$\frac{p(p-1)\cdots(p-k+1)(p-k-1)\cdots(p-n)}{k!(n-k)!}$$

Also, noting that $dx = h dp$. The limits of the integral in Equation

$$c_k = \int_a^b L_k(x) dx$$

change from 0 to n and equation reduces to

$$c_k = \frac{(-1)^{n-k} (h)}{k!(n-k)!} \int_0^n \frac{p(p-1) \cdots (p-k+1)(p-k-1) \cdots (p-n) dp}{\cdots (p-k+1)(p-k-1) \cdots (p-n) dp}$$

The error in approximating the integral can be obtained from

$$E_n = \frac{h^{n+2}}{(n+1)!} \int_0^n \frac{p(p-1) \cdots (p-n)}{(\xi)} y^{(n+1)} dp$$

Where $x_0 < \xi < x_n$. For illustration, consider the cases for $n = 1, 2$; For which we get

$$c_0 = -h \int_0^1 (p - 1) dp = \frac{h}{2},$$

$$c_1 = h \int_0^1 p dp = \frac{h}{2}$$

and

$$\begin{aligned} E_1 &= \frac{h^3}{2} y''(\xi) \int_0^1 p(p-1) dp \\ &= -\frac{h^3}{12} y''(\xi) \end{aligned}$$

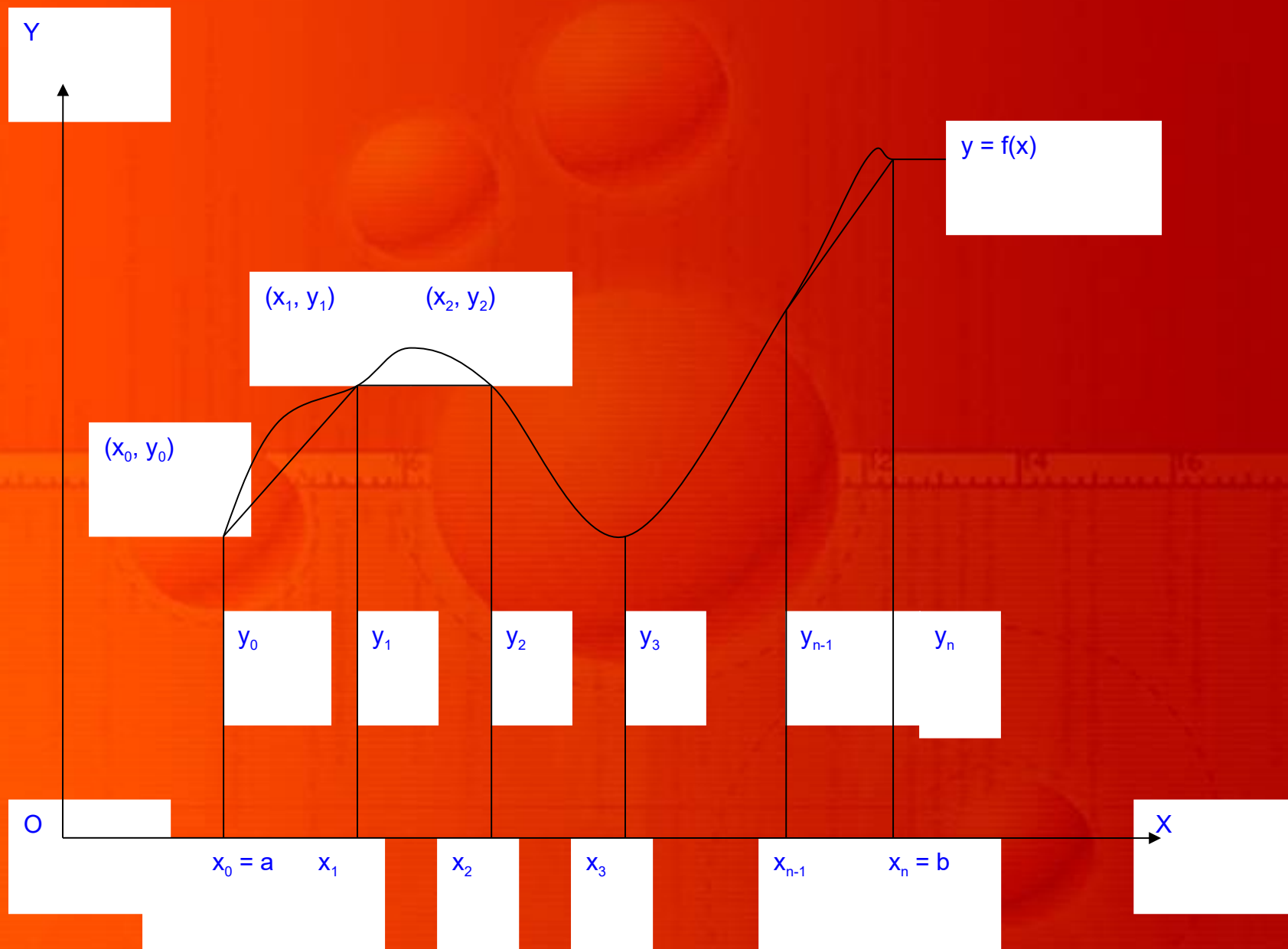
**Thus, the integration formula
is found to be**

$$\int_{x_0}^{x_1} f(x) dx = c_0 y_0 + c_1 y_1 + \text{Error}$$

$$= \frac{h}{2} (y_0 + y_1) - \frac{h^3}{2} y''(\xi)$$

This equation represents the Trapezoidal rule in the interval $[x_0, x_1]$ with error term. Geometrically, it represents an area between the curve $y = f(x)$, the x -axis and the ordinates erected

at $x = x_0$ ($= a$) and $x = x_1$ as shown in the figure.



This area is approximated by the trapezium formed by replacing the curve with its secant line drawn between the end points (x_0, y_0) and (x_1, y_1) .

For n =2, We have

$$c_0 = \frac{h}{2} \int_0^2 (p-1)(p-2) dp = \frac{h}{3}$$

$$c_1 = -h \int_0^2 p(p-2)dp = \frac{4}{3}h$$

$$c_2 = \frac{h}{2} \int_0^2 p(p-1)dp = \frac{h}{3}$$

and the error term is given by

$$E_2 = \frac{h^5}{90} y^{(iv)}(\xi)$$

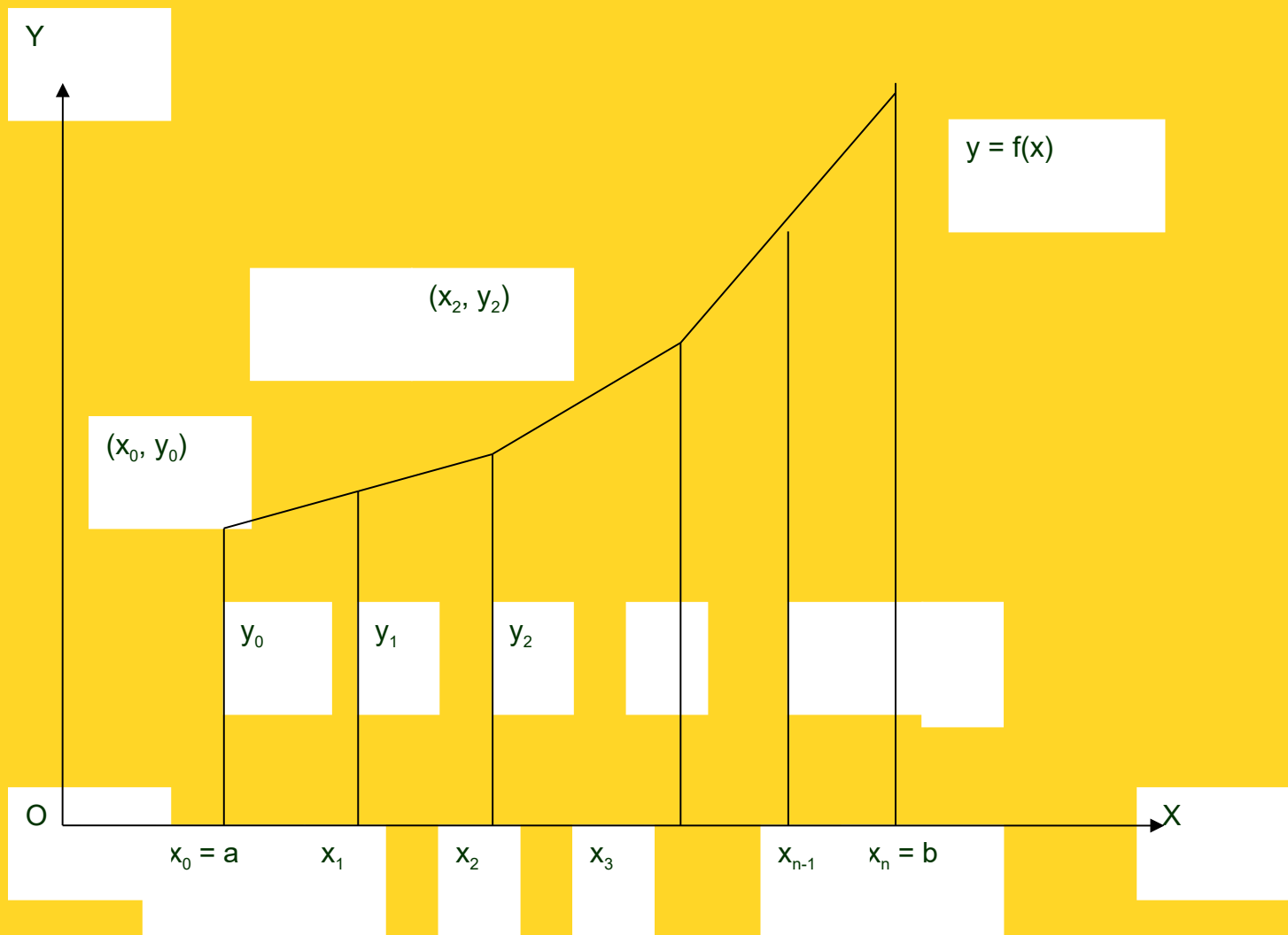
Thus, for $n = 2$, the integration takes the form

$$\int_{x_0}^{x_2} f(x) dx$$

$$= x_0 y_0 + x_1 y_1 + x_2 y_2 + \text{Error}$$

$$= \frac{h}{3} (y_0 + 4y_1 + y_2) - \frac{h^5}{90} y^{(iv)}(\xi)$$

This is known as Simpson's 1/3 rule. Geometrically, this equation represents the area between the curve $y = f(x)$, the x-axis and the ordinates at $x = x_0$ and x_2 after replacing the arc of the curve between (x_0, y_0) and (x_2, y_2) by an arc of a quadratic polynomial as in the figure



Thus Simpson's $1/3$ rule is based on fitting three points with a quadratic.

Similarly, for $n = 3$, the integration is found to be

$$\int_{x_0}^{x_3} f(x) dx$$

$$= \frac{3}{8} h (y_0 + 3y_1 + 3y_2 + y_3)$$

$$- \frac{3}{80} h^5 y^{(iv)}(\xi)$$

This is known as Simpson's 3/8 rule, which is based on fitting four points by a cubic. Still higher order Newton-Cotes integration formulae can be derived for large values of n .

**But for all practical purposes,
Simpson's $1/3$ rule is found to
be sufficiently accurate.**

Numerical

Analysis

Lecture 31

Numerical Analysis

Lecture 32

Chapter 7

Numerical Differentiation and Integration

INTRODUCTION

**DIFFERENTIATION USING
DIFFERENCE OPERATORS**

**DIFFERENTIATION USING
INTERPOLATION**

**RICHARDSON'S
EXTRAPOLATION METHOD**

NUMERICAL INTEGRATION

NEWTON-COTES
INTEGRATION FORMULAE
THE TRAPEZOIDAL RULE
(COMPOSITE FORM)
SIMPSON'S RULES
(COMPOSITE FORM)
ROMBERG'S INTEGRATION
DOUBLE INTEGRATION

Basic Issues in Integration

What does an integral represent?

$$\int_a^b f(x) dx = \textit{AREA}$$

$$\int_c^d \int_a^b g(x, y) dx dy = \textit{VOLUME}$$

Basic definition of an integral:

$$\int_a^b f(x) dx =$$

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n f(x_k) \Delta x$$

sum of Height x Width

Objective:

Evaluate $I = \int_a^b f(x) dx$ without

doing calculation analytically.

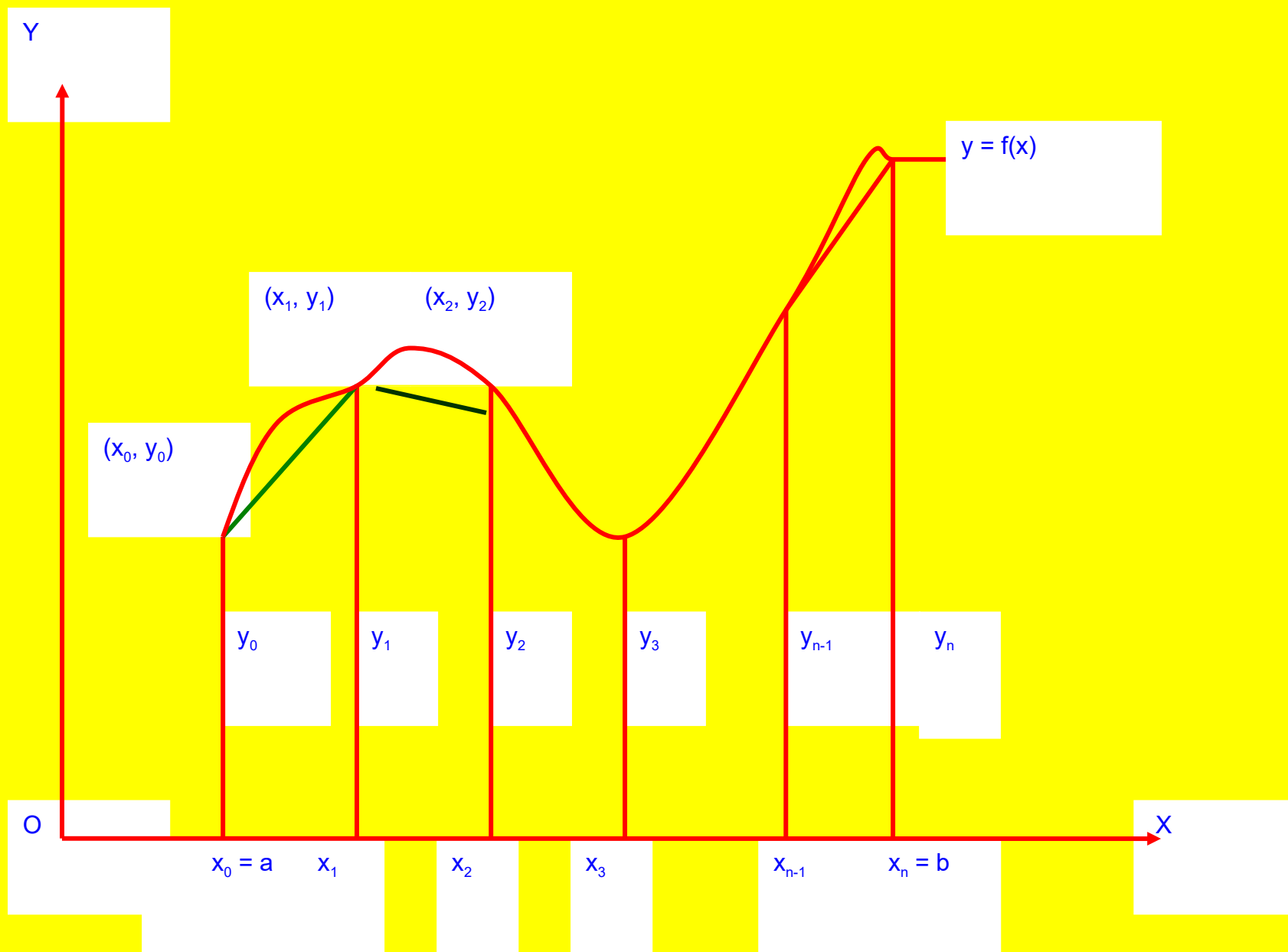
When would we want to do this?

NUMERICAL INTEGRATION

Consider the definite integral

$$I = \int_{x=a}^b f(x) dx$$

where $f(x)$ is known either explicitly or is given as a table of values corresponding to some values of x , whether equispaced or not. Integration of such functions can be carried out using numerical techniques.



This area is approximated by the trapezium formed by replacing the curve with its secant line drawn between the end points (x_0, y_0) and (x_1, y_1) .

$$\int_{x_0}^{x_1} f(x) dx = c_0 y_0 + c_1 y_1 + \text{Error}$$

$$= \frac{h}{2} (y_0 + y_1) - \frac{h^3}{2} y''(\xi)$$

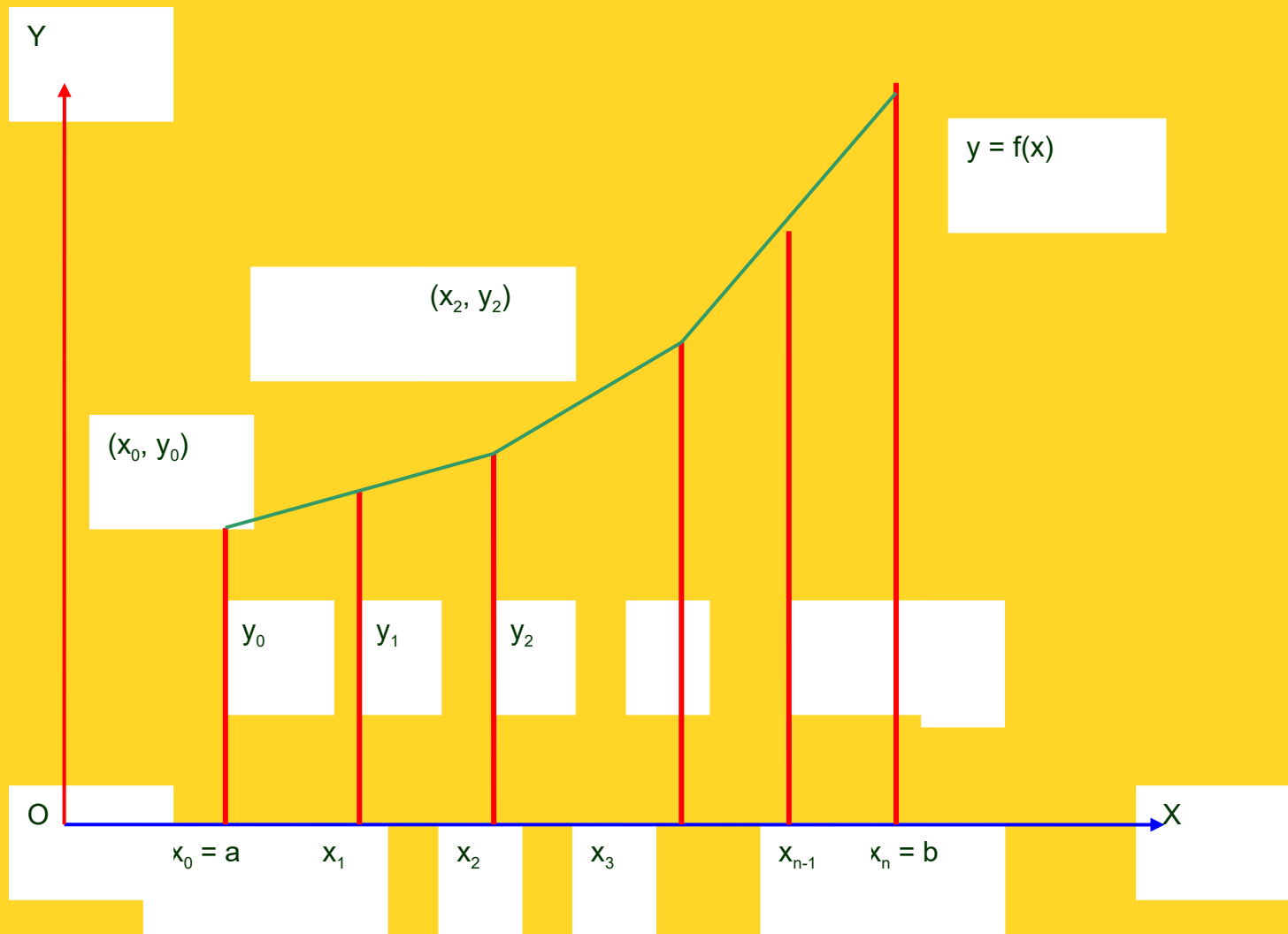
Then, if $n = 2$, the integration takes the form

$$\int_{x_0}^{x_2} f(x) dx$$

$$= x_0 y_0 + x_1 y_1 + x_2 y_2 + \text{Error}$$

$$= \frac{h}{3} (y_0 + 4y_1 + y_2) - \frac{h^5}{90} y^{(iv)}(\xi)$$

This is known as Simpson's 1/3 rule. Geometrically, this equation represents the area between the curve $y = f(x)$, the x-axis and the ordinates at $x = x_0$ and x_2 after replacing the arc of the curve between (x_0, y_0) and (x_2, y_2) by an arc of a quadratic polynomial as in the figure



Thus Simpson's $1/3$ rule is based on fitting three points with a quadratic.

Similarly, for $n = 3$, the integration is found to be

$$\int_{x_0}^{x_3} f(x) dx$$

$$= \frac{3}{8} h (y_0 + 3y_1 + 3y_2 + y_3)$$

$$- \frac{3}{80} h^5 y^{(iv)}(\xi)$$

This is known as Simpson's 3/8 rule, which is based on fitting four points by a cubic. Still higher order Newton-Cotes integration formulae can be derived for large values of n .

**But for all practical purposes,
Simpson's $1/3$ rule is found to
be sufficiently accurate.**

The Trapezoidal Rule (Composite Form)

**The Newton-Cotes formula is
based on approximating
 $y = f(x)$ between (x_0, y_0) and
 (x_1, y_1)**

**by a straight line, thus
forming a trapezium,
is called trapezoidal rule. In
order to evaluate the definite
integral**

$$I = \int_a^b f(x) dx$$

we divide the interval $[a, b]$ into n sub-intervals, each of size $h = (b - a)/n$ and denote the sub-intervals by $[x_0, x_1], [x_1, x_2], \dots, [x_{n-1}, x_n]$, such that $x_0 = a$ and $x_n = b$ and

$$x_k = x_0 + k_h, k = 1, 2, \dots, n - 1.$$

Thus, we can write the above definite integral as a sum.

Therefore,

$$I = \int_{x_0}^{x_n} f(x) dx$$

$$= \int_{x_0}^{x_1} f(x) dx + \int_{x_1}^{x_2} f(x) dx$$

$$+ \cdots + \int_{x_{n-1}}^{x_n} f(x) dx$$

The area under the curve in each sub-interval is approximated by a trapezium. The integral I , which represents an area between the curve $y = f(x)$, the x-axis and the ordinates at $x = x_0$

and $x = x_n$ is obtained by adding all the trapezoidal areas in each sub-interval.

Now, using the trapezoidal rule into equation:

$$\int_{x_0}^{x_1} f(x) dx = c_0 y_0 + c_1 y_1 + \text{Error}$$

$$= \frac{h}{2}(y_0 + y_1) - \frac{h^3}{2} y''(\xi)$$

we get

$$I = \int_{x_0}^{x_n} f(x) dx$$

$$= \frac{h}{2}(y_0 + y_1) - \frac{h^3}{2} y''(\xi_1)$$

$$+ \frac{h}{2}(y_1 + y_2) - \frac{h^3}{12} y''(\xi_2)$$

$$+ \cdots + \frac{h}{2}(y_{n-1} + y_n) - \frac{h^3}{12} y''(\xi_n)$$

where $x_{k-1} < \xi < x_k$, for $k = 1, 2, \dots, n-1$.

Thus, we arrive at the result

$$\int_{x_0}^{x_n} f(x) dx$$

$$= \frac{h}{2} (y_0 + 2y_1 + 2y_2$$

$$+ \dots + 2y_{n-1} + y_n) + E_n$$

where the error term E_n is given by

$$E_n = -\frac{h^3}{12} [y''(\xi_1) + y''(\xi_2) + \cdots + y''(\xi_n)]$$

Equation represents the trapezoidal rule over $[x_0, x_n]$, which is also called the composite form of the trapezoidal rule.

The error term given by Equation:

$$E_n = -\frac{h^3}{12} [y''(\xi_1) + y''(\xi_2) + \cdots + y''(\xi_n)]$$

is called the global error.

However, if we assume

that $y''(x)$ is

continuous over $[x_0, x_n]$ then

there exists some ξ in $[x_0, x_n]$

such that $x_n = x_0 + nh$

and

$$E_n = -\frac{h^3}{12} [ny''(\xi)]$$

$$= -\frac{x_n - x_0}{12} h^2 y''(\xi)$$

Then the global error can be conveniently written as $O(h^2)$.

Simpson's Rules (Composite Forms)

In deriving equation. ,

$$\int_{x_0}^{x_2} f(x) dx$$

$$= x_0 y_0 + x_1 y_1 + x_2 y_2 + \text{Error}$$

$$= \frac{h}{3} (y_0 + 4y_1 + y_2) - \frac{h^5}{90} y^{(iv)}(\xi)$$

the Simpson's $1/3$ rule, we have used two sub-intervals of equal width. In order to get a composite formula, we shall divide the interval of integration $[a, b]$ into an even number of sub-intervals say $2N$,

each of width $(b - a)/2N$,
thereby we have

$$x_0 = a, x_1, \dots, x_{2N} = b \quad \text{and}$$

$$x_k = x_0 + kh, k = 1, 2, \dots (2N - 1).$$

Thus, the definite integral
I can be written as

$$I = \int_a^b f(x) dx$$

$$= \int_{x_0}^{x_2} f(x) dx + \int_{x_2}^{x_4} f(x) dx$$

$$+ \cdots + \int_{x_{2N-2}}^{x_{2N}} f(x) dx$$

Applying Simpson's 1/3 rule as in equation

$$\int_{x_0}^{x_2} f(x) dx$$

$$= x_0 y_0 + x_1 y_1 + x_2 y_2 + \text{Error}$$

$$= \frac{h}{3} (y_0 + 4y_1 + y_2) - \frac{h^5}{90} y^{(iv)}(\xi)$$

to each of the integrals on the right-hand side of the above equation, we obtain

$$I = \frac{h}{3} [(y_0 + 4y_1 + y_2) + (y_2 + 4y_3 + y_4) + \cdots]$$

$$+ (y_{2N-2} + 4y_{2N-1} + y_{2N})]$$

$$-\frac{N}{90}h^5y^{(iv)}(\xi)$$

That is,

$$\int_{x_0}^{x_{2N}} f(x) dx$$

$$\begin{aligned} &= \frac{h}{3} [y_0 + 4(y_1 + y_3 + \cdots + y_{2N-1}) \\ &\quad + 2(y_2 + y_4 + \cdots + y_{2N-2}) + y_{2N}] \\ &\quad + \text{Error term} \end{aligned}$$

This formula is called composite Simpson's 1/3 rule. The error term E , which is also called global error, is given by

$$E = -\frac{N}{90} h^5 y^{(iv)}(\xi)$$

$$= -\frac{x_{2N} - x_0}{180} h^4 y^{(iv)}(\xi)$$

for some ξ in $[x_0, x_{2N}]$. Thus, in Simpson's 1/3 rule, the global error is of $O(h^2)$.

Similarly in deriving composite Simpson's 3/8 rule, we divide the interval of integration into n sub-intervals, where n is divisible by 3,

**and applying the integration
formula**

$$\int_{x_0}^{x_3} f(x) dx$$

$$= \frac{3}{8} h (y_0 + 3y_1 + 3y_2 + y_3)$$

$$- \frac{3}{80} h^5 y^{(iv)}(\xi)$$

**to each of the integral given
below**

$$\int_{x_0}^{x_n} f(x) dx$$

$$= \int_{x_0}^{x_3} f(x) dx + \int_{x_3}^{x_6} f(x) dx + \cdots$$

$$+ \int_{x_{n-3}}^{x_n} f(x) dx$$

**we obtain the composite form
of Simpson's 3/8 rule as**

$$\int_a^b f(x) dx$$

$$= \frac{3}{8} h [y(a) + 3y_1 + 3y_2 + 2y_3$$

$$+ 3y_4 + 3y_5 + 2y_6 + \dots$$

$$+ 2y_{n-3} + 3y_{n-2} + 3y_{n-1} + y(b)]$$

with the global error E given by

$$E = -\frac{x_n - x_0}{80} h^4 y^{(iv)}(\xi)$$

It may be noted that the global error in Simpson's $1/3$ and $3/8$ rules are of the same order. However, if we consider the magnitudes of the error terms, we notice that Simpson's $1/3$ rule is superior to Simpson's $3/8$ rule. For illustration, we consider few examples.

Example

Find the approximate value of

$$y = \int_0^{\pi} \sin x dx$$

using

(i) trapezoidal rule

(ii) **Simpson's $1/3$ rule** by dividing the range of integration into six equal parts. Calculate the percentage error from its true value in both the cases.

Solution We shall at first divide the range of integration $(0, \pi)$ into six equal parts so that each part is of width $\pi/6$ and write down the table of values:

x **$y = \sin x$**

0 **0.0**

$\pi/6$ **0.5**

$\pi/3$ **0.8660**

$\pi/2$ **1.0**

$2\pi/3$ **0.8660**

$5\pi/6$ **0.5**

π **0.**

Applying trapezoidal rule, we have

$$\int_0^{\pi} \sin x dx$$

$$= \frac{h}{2} [y_0 + y_6 + 2(y_1 + y_2 + y_3 + y_4 + y_5)]$$

Here, h , the width of the interval is $\pi / 6$. Therefore,

$$y = \int_0^{\pi} \sin x dx$$

$$= \frac{\pi}{12} [0 + 0 + 2(3.732)]$$

$$= \frac{3.1415}{6} \times 3.732 = 1.9540$$

Applying Simpson's 1/3 rule

$$\int_{x_0}^{x_2} f(x) dx$$

$$= x_0 y_0 + x_1 y_1 + x_2 y_2 + \text{Error}$$

$$= \frac{h}{3} (y_0 + 4y_1 + y_2) - \frac{h^5}{90} y^{(iv)}(\xi)$$

We have

$$\int_0^{\pi} \sin x dx$$

$$= \frac{h}{3} [y_0 + y_6 + 4(y_1 + y_3 + y_5) + 2(y_2 + y_4)]$$

$$= \frac{\pi}{18} [0 + 0 + (4 \times 2) + (2)(1.732)]$$

$$= \frac{3.1415}{18} \times 11.464 = 2.0008$$

But the actual value of the integral is

$$\int_0^{\pi} \sin x dx = [-\cos x]_0^{\pi} = 2$$

Hence, in the case of trapezoidal rule

The percentage of error

$$= \frac{2 - 1.954}{2} \times 100 = 23$$

While in the case of Simpson's rule the percentage error is

$$\frac{2 - 2.0008}{2} \times 100 = 0.04$$

(sign ignored)

Numerical Analysis

Lecture 32

Numerical Analysis

Lecture 33

Chapter 7

Numerical Differentiation and Integration

INTRODUCTION

**DIFFERENTIATION USING
DIFFERENCE OPERATORS**

**DIFFERENTIATION USING
INTERPOLATION**

**RICHARDSON'S
EXTRAPOLATION METHOD**

NUMERICAL INTEGRATION

NEWTON-COTES
INTEGRATION FORMULAE
THE TRAPEZOIDAL RULE
(COMPOSITE FORM)
SIMPSON'S RULES
(COMPOSITE FORM)
ROMBERG'S INTEGRATION
DOUBLE INTEGRATION

Basic Issues in Integration

What does an integral represent?

$$\int_a^b f(x) dx = \textit{AREA}$$

$$\int_c^d \int_a^b g(x, y) dx dy = \textit{VOLUME}$$

Basic definition of an integral:

$$\int_a^b f(x) dx =$$

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n f(x_k) \Delta x$$

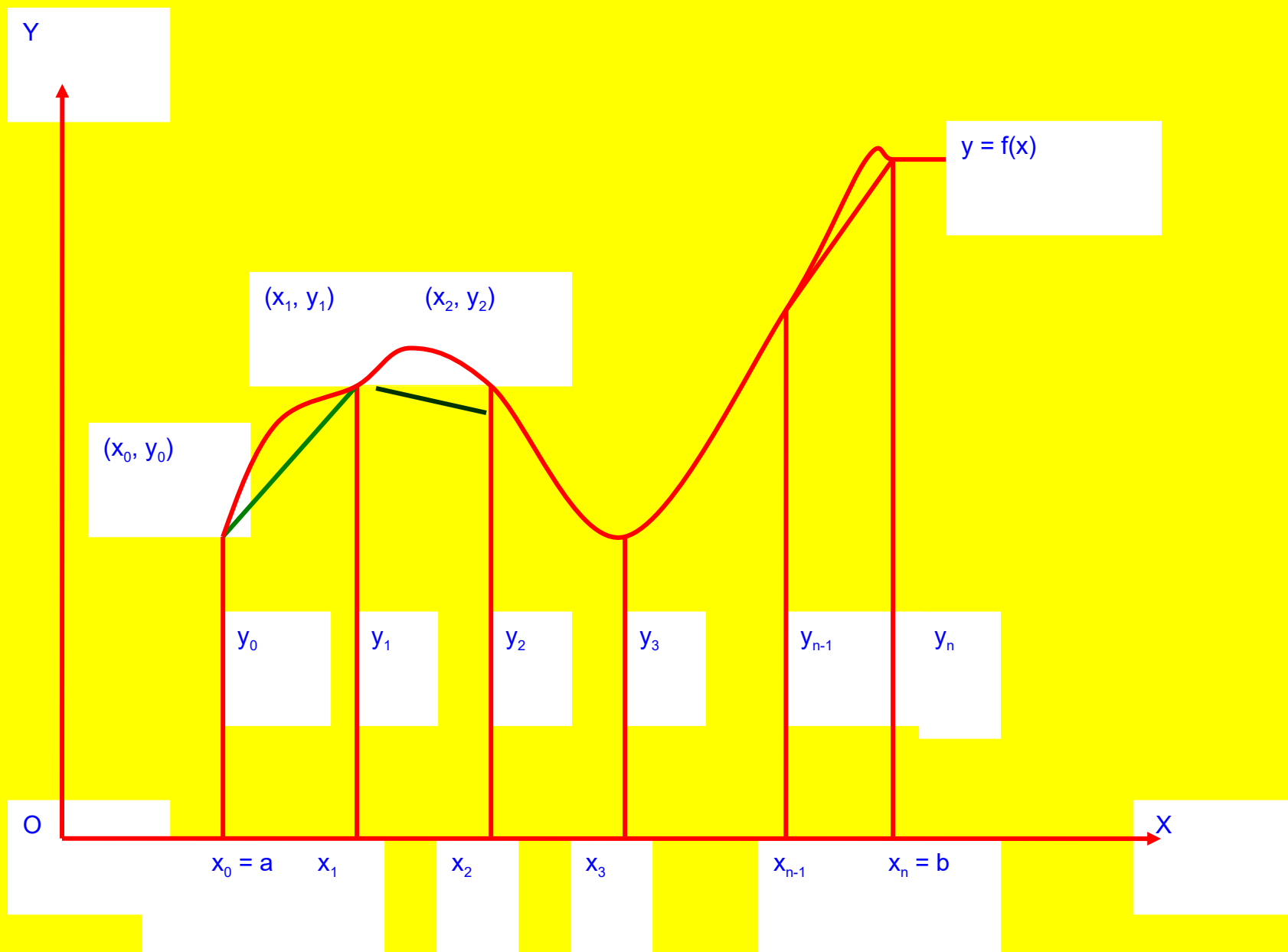
sum of Height x Width

NUMERICAL INTEGRATION

Consider the definite integral

$$I = \int_{x=a}^b f(x) dx$$

where $f(x)$ is known either explicitly or is given as a table of values corresponding to some values of x , whether equispaced or not. Integration of such functions can be carried out using numerical techniques.



This area is approximated by the trapezium formed by replacing the curve with its secant line drawn between the end points (x_0, y_0) and (x_1, y_1) .

$$\int_{x_0}^{x_1} f(x) dx = c_0 y_0 + c_1 y_1 + \text{Error}$$

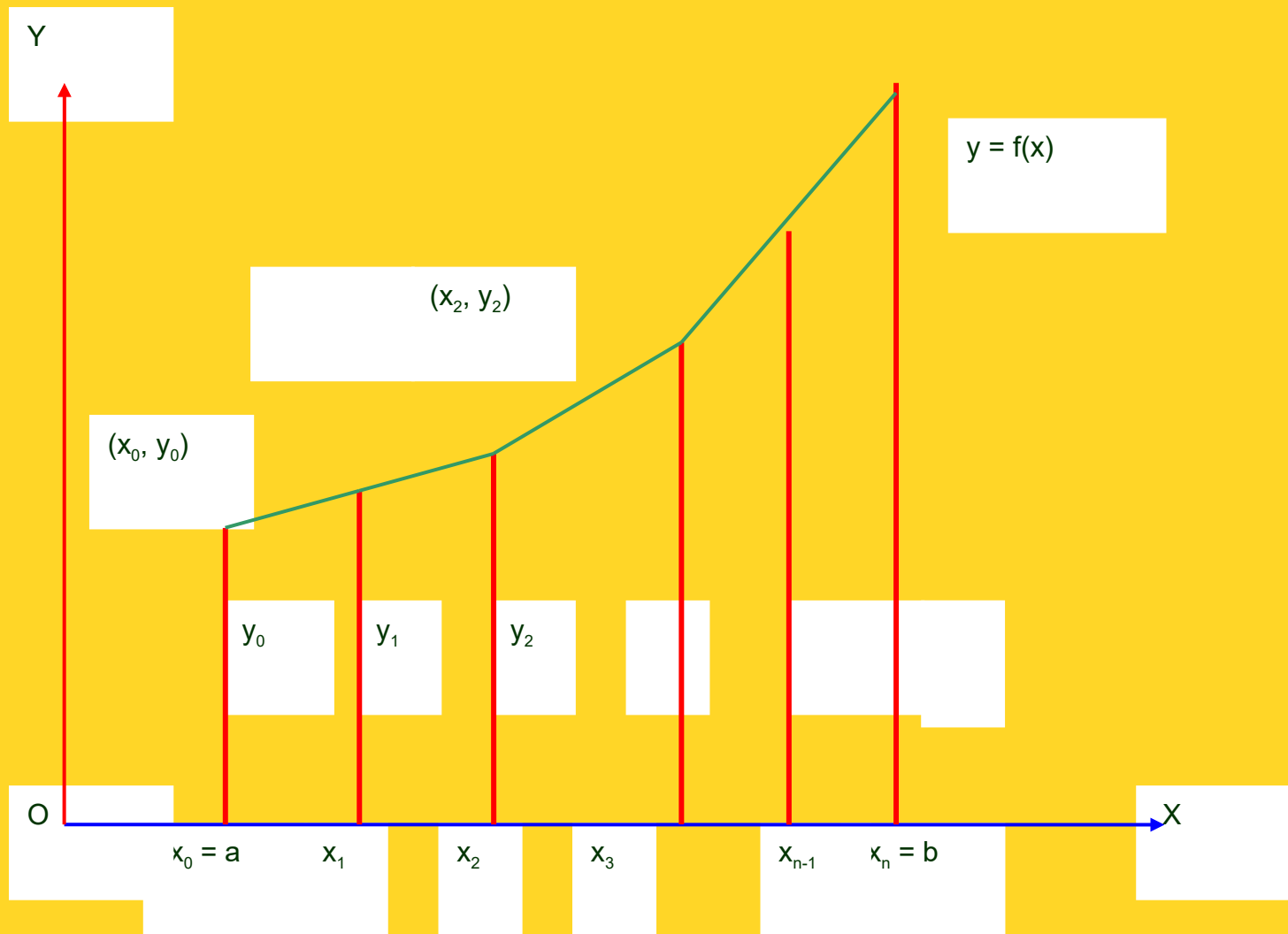
$$= \frac{h}{2} (y_0 + y_1) - \frac{h^3}{12} y''(\xi)$$

Then, if $n = 2$, the integration takes the form

$$\int_{x_0}^{x_2} f(x) dx$$

$$= x_0 y_0 + x_1 y_1 + x_2 y_2 + \text{Error}$$

$$= \frac{h}{3} (y_0 + 4y_1 + y_2) - \frac{h^5}{90} y^{(iv)}(\xi)$$



Thus Simpson's $1/3$ rule is based on fitting three points with a quadratic.

Similarly, for $n = 3$, the integration is found to be

$$\int_{x_0}^{x_3} f(x) dx$$

$$= \frac{3}{8} h (y_0 + 3y_1 + 3y_2 + y_3)$$

$$- \frac{3}{80} h^5 y^{(iv)}(\xi)$$

This is known as Simpson's 3/8 rule, which is based on fitting four points by a cubic. Still higher order Newton-Cotes integration formulae can be derived for large values of n .

The Trapezoidal Rule (Composite Form)

**The Newton-Cotes formula is
based on approximating
 $y = f(x)$ between (x_0, y_0) and
 (x_1, y_1)**

TRAPEZOIDAL RULE

$$\int_{x_0}^{x_n} f(x) dx$$

$$= \frac{h}{2} (y_0 + 2y_1 + 2y_2$$

$$+ \cdots + 2y_{n-1} + y_n) + E_n$$

$$\int_{x_0}^{x_n} f(x) dx$$

$$= \frac{h}{2} (y_0 + 2y_1 + 2y_2$$

$$+ \cdots + 2y_{n-1} + y_n) + E_n$$

SIMPSON'S 1/3 RULE

$$I = \int_{x_0}^{x_2} f(x) dx$$

$$= \frac{h}{3} (y_0 + 4y_1 + y_2) - \frac{h^5}{90} y^{(iv)}(\xi)$$

$$\int_{x_0}^{x_{2N}} f(x) dx$$

$$\begin{aligned} &= \frac{h}{3} [y_0 + 4(y_1 + y_3 + \cdots + y_{2N-1}) \\ &\quad + 2(y_2 + y_4 + \cdots + y_{2N-2}) + y_{2N}] \\ &\quad + \text{Error term} \end{aligned}$$

$$E = -\frac{x_{2N} - x_0}{180} h^4 y^{(iv)}(\xi)$$

Simpson's 3/8 rule is

$$\int_a^b f(x)dx$$

$$= \frac{3}{8}h[y(a) + 3y_1 + 3y_2 + 2y_3$$

$$+ 3y_4 + 3y_5 + 2y_6 + \dots$$

$$+ 2y_{n-3} + 3y_{n-2} + 3y_{n-1} + y(b)]$$

with the global error E given by

$$E = -\frac{x_n - x_0}{80} h^4 y^{(iv)}(\xi)$$

Example

Find the approximate value of

$$y = \int_0^{\pi} \sin x dx$$

using

(i) trapezoidal rule

(ii) Simpson's $1/3$ rule by dividing the range of integration into six equal parts. Calculate the percentage error from its true value in both the cases.

Solution We shall at first divide the range of integration $(0, \pi)$ into six equal parts so that each part is of width $\pi/6$ and write down the table of values:

x **$y = \sin x$**

0 **0.0**

$\pi/6$ **0.5**

$\pi/3$ **0.8660**

$\pi/2$ **1.0**

$2\pi/3$ **0.8660**

$5\pi/6$ **0.5**

π **0.**

Applying trapezoidal rule, we have

$$\int_0^{\pi} \sin x dx$$

$$= \frac{h}{2} [y_0 + y_6 + 2(y_1 + y_2 + y_3 + y_4 + y_5)]$$

Here, h, the width of the interval is $\pi /6$. Therefore,

$$y = \int_0^{\pi} \sin x dx$$

$$= \frac{\pi}{12} [0 + 0 + 2(3.732)]$$

$$= \frac{3.1415}{6} \times 3.732 = 1.9540$$

Applying Simpson's 1/3 rule

$$\int_{x_0}^{x_2} f(x) dx$$

$$= x_0 y_0 + x_1 y_1 + x_2 y_2 + \text{Error}$$

$$= \frac{h}{3} (y_0 + 4y_1 + y_2) - \frac{h^5}{90} y^{(iv)}(\xi)$$

We have

$$\int_0^{\pi} \sin x dx$$

$$= \frac{h}{3} [y_0 + y_6 + 4(y_1 + y_3 + y_5) + 2(y_2 + y_4)]$$

$$= \frac{\pi}{18} [0 + 0 + (4 \times 2) + (2)(1.732)]$$

$$= \frac{3.1415}{18} \times 11.464 = 2.0008$$

But the actual value of the integral is

$$\int_0^{\pi} \sin x \, dx = [-\cos x]_0^{\pi} = 2$$

Hence, in the case of trapezoidal rule

The percentage of error

$$= \frac{2 - 1.954}{2} \times 100 = 23$$

While in the case of Simpson's 1/3rd rule the percentage error is

$$\frac{2 - 2.0008}{2} \times 100 = 0.04$$

(sign ignored)

Example : From the following data, estimate the value of

$$\int_1^5 \log x dx .$$

using Simpson's 1/3 rule.

Also, obtain the value of h, so

**that the value of the integral
will be accurate up to five
decimal places.**

x	$y = \log x$
1.0	0.0000
1.5	0.4055
2.0	0.6931
2.5	0.9163
3.0	1.0986
3.5	1.2528
4.0	1.3863
4.5	1.5041
5.0	1.6094

Solution We have from the data, $n = 0, 1, \dots, 8$, and $h = 0.5$. Now using Simpson's 1/3 rule,

$$\int_1^5 \log x dx$$

$$= \frac{h}{3} [y_0 + y_8 + 4(y_1 + y_3 + y_5 + y_7) + 2(y_2 + y_4 + y_6)]$$

$$= \frac{0.5}{3} [(0 + 1.6094) + 4(4.0787) + 2(3.178)]$$

$$= \frac{0.5}{3} (1.6094 + 16.3148 + 6.356)$$

$$= 4.0467$$

The error in Simpson's rule is given by

$$E = -\frac{x_{2N} - x_0}{180} h^4 y^{(iv)}(\xi)$$

(ignoring the sign)

Since

$$y = \log x, y' = \frac{1}{x},$$

$$y'' = -\frac{1}{x^2}, y''' = \frac{2}{x^3},$$

$$y^{(iv)} = -\frac{6}{x^4}$$

$$\mathop{Max}_{1 \leq x \leq 5} y^{(iv)}(x) = 6,$$

$$\mathop{Max}_{1 \leq x \leq 5} y^{(iv)}(x) = 0.0096.$$

Therefore, the error bounds are given by

$$\frac{(0.0096)(4)h^4}{180} < E < \frac{(6)(4)h^4}{180}$$

If the result is to be accurate up to five decimal places, then

$$\frac{24h^4}{180} < 10^{-5}$$

That is, $h^4 < 0.000075$ or $h < 0.09$. It may be noted that the actual value of integrals is

$$\int_1^5 \log x dx = [x \log x - x]_1^5$$
$$= 5 \log 5 - 4$$

Example : Evaluate the integral

$$I = \int_0^1 \frac{dx}{1+x^2}$$

by using (i) trapezoidal rule

(ii) Simpson's $1/3$ rule by taking $h = 1/4$. Hence compute the approximate value of π .

**Solution At first, we shall
tabulate the function as**

X	$y = 1 / 1+x^2$

0	1

1/4	0.9412

1/2	0.8000

3/4	0.6400

1	0.5000

**using trapezoidal rule, and taking
 $h = 1/4$**

$$I = \frac{h}{2} [y_0 + y_4 + 4(y_1 + y_2 + y_3)]$$
$$= \frac{1}{8} [1.5 + 2(2.312)] = 0.7828$$

using Simpson's 1/3 rule, and taking $h = 1/4$, we have

$$I = \frac{h}{3} [y_0 + y_4 + 4(y_1 + y_3) + 2y_2]$$
$$= \frac{1}{12} [1.5 + 4(1.512) + 1.6] = 0.7854$$

But the closed form solution to the given integral is

$$\int_0^1 \frac{dx}{1+x^2} + \left[\tan^{-1} x \right]_0^1 = \frac{\pi}{4}$$

**Equating the last two
equations , we get $\pi = 3.1416$.**

Example : Compute the integral

$$I = \sqrt{\frac{2}{\pi}} \int_0^1 e^{-x^2/2} dx$$

**using Simpson's 1/3 rule,
taking $h = 0.125$.**

Solution **At the outset, we shall construct the table of the function as required.**

x

$$y = \sqrt{\frac{2}{\pi}} \exp(-x^2 / 2)$$

0 0.7979

0.125 0.7917

0.250 0.7733

0.375 0.7437

0.5 0.7041

0.625 0.6563

0.750 0.6023

0.875 0.5441

1.0 0.4839

Using Simpson's 1.3 rule, we have

$$I = \frac{h}{3} [y_0 + y_8 + 4(y_1 + y_3 + y_5 + y_7)$$

$$+ 2(y_2 + y_4 + y_6)]$$

$$= \frac{0.125}{3} [0.7979 + 0.4839$$

$$+ 4(0.7917 + 0.7437 + 0.6563 + 0.5441)]$$

$$\begin{aligned}& + 2(0.7733 + 0.7041 + 0.6023)] \\&= \frac{0.125}{3}(1.2812 + 10.9432 + 4.1594) \\&= 0.6827\end{aligned}$$

Hence, $I = 0.6827$.

Example : A missile is launched from a ground station. The acceleration during first 80 seconds of flight, as recorded, is given in the following table:

$t(s)$	$a(m / s^2)$
0	30.00
10	31.63
20	33.34
30	35.47
40	37.75
50	40.33
60	43.25
70	46.69
80	50.67

**compute the velocity of the missile when $t = 80$ s,
using Simpson's 1/3 rule.**

Solution : Since acceleration is defined as the rate of change of velocity, we have

$$v = \int_0^{80} a \, dt, \text{ or, } \frac{dv}{dt} = a$$

Using Simpson's 1/3-rule, we have

$$\begin{aligned}v &= \frac{h}{3}[(y_0 + y_8) + 4(y_1 + y_3 + y_7) + 2(y_2 + y_4 + y_6)] \\&= \frac{10}{3}[(30 + 50.67) + 4(31.63 + 35.47 + 40.33 + 46.69) \\&\quad + 2(33.34 + 37.75 + 43.25)] \\&= 3086.1 \text{ m/s}\end{aligned}$$

Therefore, the required velocity is given by $y = 3.0861$ km/s.

Numerical Analysis

Lecture 33

Numerical Analysis

Lecture 34

Chapter 7

Numerical Differentiation and Integration

INTRODUCTION

**DIFFERENTIATION USING
DIFFERENCE OPERATORS**

**DIFFERENTIATION USING
INTERPOLATION**

**RICHARDSON'S
EXTRAPOLATION METHOD**

NUMERICAL INTEGRATION

NEWTON-COTES
INTEGRATION FORMULAE
THE TRAPEZOIDAL RULE
(COMPOSITE FORM)
SIMPSON'S RULES
(COMPOSITE FORM)
ROMBERG'S INTEGRATION
DOUBLE INTEGRATION

Basic Issues in Integration

What does an integral represent?

$$\int_a^b f(x) dx = \text{AREA}$$

$$\int_c^d \int_a^b g(x, y) dx dy = \text{VOLUME}$$

NUMERICAL INTEGRATION

Consider the definite integral

$$I = \int_{x=a}^b f(x) dx$$

$$\int_{x_0}^{x_1} f(x) dx = c_0 y_0 + c_1 y_1 + \text{Error}$$

$$= \frac{h}{2} (y_0 + y_1) - \frac{h^3}{12} y''(\xi)$$

Then, if $n = 2$, the integration takes the form

$$\int_{x_0}^{x_2} f(x) dx$$

$$= x_0 y_0 + x_1 y_1 + x_2 y_2 + \text{Error}$$

$$= \frac{h}{3} (y_0 + 4y_1 + y_2) - \frac{h^5}{90} y^{(iv)}(\xi)$$

Thus Simpson's $1/3$ rule is based on fitting three points with a quadratic.

Similarly, for $n = 3$, the integration is found to be

$$\int_{x_0}^{x_3} f(x) dx$$

$$= \frac{3}{8} h (y_0 + 3y_1 + 3y_2 + y_3)$$

$$- \frac{3}{80} h^5 y^{(iv)}(\xi)$$

This is known as Simpson's 3/8 rule, which is based on fitting four points by a cubic. Still higher order Newton-Cotes integration formulae can be derived for large values of n .

TRAPEZOIDAL RULE

$$\int_{x_0}^{x_n} f(x) dx$$

$$= \frac{h}{2} (y_0 + 2y_1 + 2y_2$$

$$+ \cdots + 2y_{n-1} + y_n) + E_n$$

$$\int_{x_0}^{x_n} f(x) dx$$

$$= \frac{h}{2} (y_0 + 2y_1 + 2y_2$$

$$+ \cdots + 2y_{n-1} + y_n) + E_n$$

SIMPSON'S 1/3 RULE

$$I = \int_{x_0}^{x_2} f(x) dx$$

$$= \frac{h}{3} (y_0 + 4y_1 + y_2) - \frac{h^5}{90} y^{(iv)}(\xi)$$

$$\int_{x_0}^{x_{2N}} f(x) dx$$

$$\begin{aligned} &= \frac{h}{3} [y_0 + 4(y_1 + y_3 + \cdots + y_{2N-1}) \\ &\quad + 2(y_2 + y_4 + \cdots + y_{2N-2}) + y_{2N}] \\ &\quad + \text{Error term} \end{aligned}$$

$$E = -\frac{x_{2N} - x_0}{180} h^4 y^{(iv)}(\xi)$$

Simpson's 3/8 rule is

$$\int_a^b f(x) dx$$

$$= \frac{3}{8} h [y(a) + 3y_1 + 3y_2 + 2y_3$$

$$+ 3y_4 + 3y_5 + 2y_6 + \cdots$$

$$+ 2y_{n-3} + 3y_{n-2} + 3y_{n-1} + y(b)]$$

with the global error E given by

$$E = -\frac{x_n - x_0}{80} h^4 y^{(iv)}(\xi)$$

ROMBERG'S INTEGRATION

We have observed that the trapezoidal rule of integration of a definite integral is of $O(h^2)$, while that of Simpson's $1/3$ and $3/8$ rules are of fourth-order accurate.

We can improve the accuracy of trapezoidal and Simpson's rules using Richardson's extrapolation procedure which is also called Romberg's integration method.

**For example, the error in
trapezoidal rule of a definite
integral**

$$I = \int_a^b f(x) dx$$

can be written in the form

$$I = I_T + c_1 h^2 + c_2 h^4 + c_3 h^6 + \dots$$

By applying Richardson's extrapolation procedure to trapezoidal rule, we obtain the following general formula

$$I_{Tm} \left(\frac{h}{2^m} \right)$$

$$= \frac{4^m I_{T(m-1)} \left(\frac{h}{2^m} \right) - I_{T(m-1)} \left(\frac{h}{2^{m-1}} \right)}{4^{m-1}}$$

**where $m = 1, 2, \dots$,
with**

$$I_{T_0}(h) = I_T(h).$$

**For illustration, we consider
the following example.**

Example: Using Romberg's integration method, find the value of

$$\int_1^{1.8} y(x) dx$$

**starting with trapezoidal rule,
for the tabular values**

x	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8
y = f(x)	1.543	1.669	1.811	1.971	2.151	2.352	2.577	2.828	3.107

Solution Taking

$$x_0 = 1, \quad x_n = 1.8,$$

$$h = \frac{1.8 - 1.0}{N}, \quad x_i = x_0 + ih$$

Let I_T denote the integration by Trapezoidal rule, then for

$$\begin{aligned} N = 1, h = 0.8, I_T &= \frac{h}{2} (y_0 + y_1) \\ &= 0.4(1.543 + 3.107) = 1.8600 \end{aligned}$$

$$N = 2, h = 0.4, I_T$$

$$= \frac{h}{2} (y_0 + 2y_1 + y_2)$$

$$= 0.2[1.543 + 2(2.151) \\ + 3.107]$$

$$= 1.7904$$

$$N = 4, h = 0.2, I_T$$

$$= \frac{h}{2} [y_0 + 2(y_1 + y_2 + y_3) + y_4]$$

$$= 0.1 [1.543 + 2(1.811 + 2.151 + 2.577) + 3.107]$$

$$= 1.7728$$

Similarly for

$$N = 8, \quad h = 0.1,$$

$$I_T = 1.7684$$

Now, using Romberg's formula , we have

$$I_{T1}\left(\frac{h}{2}\right) = \frac{4(1.7904) - 1.8600}{3}$$
$$= 1.7672$$

$$I_{T\ 2}\left(\frac{h}{2^2}\right) = \frac{4^2(1.7728) - 1.7672}{4^2 - 1}$$

$$= 1.77317$$

$$I_{T3} \left(\frac{h}{2^3} \right) = \frac{4^3 (1.7672) - 1.77317}{4^3 - 1}$$
$$= 1.7671$$

Thus, after three steps, it is found that the value of the tabulated integral is 1.7671.

DOUBLE INTEGRATION

To evaluate numerically a double integral of the form

$$I = \int \left[\int (x, y) dx \right] dy$$

**over a rectangular region
bounded by the lines $x = a$, $x = b$, $y = c$, $y = d$ we shall
employ either trapezoidal rule
or Simpson's rule, repeatedly
With respect to one variable at
a time.**

Noting that, both the integrations are just a linear combination of values of the given function at different values of the independent variable, we divide the interval $[a, b]$ into N equal

sub-intervals of size h , such that $h = (b - a)/N$; and the interval (c, d) into M equal sub-intervals of size k , so that $k = (d - c)/M$. Thus, we have

$$x_i = x_0 + ih, \quad x_0 = a,$$

$$x_N = b, \quad \text{for } i = 1, 2, \dots, N - 1$$

$$y_i = y_0 + ik, \quad y_0 = c,$$

$$y_M = d, \quad \text{for } i = 1, 2, \dots, M - 1$$

Thus, we can generate a table of values of the integrand, and the above procedure of integration is illustrated by considering a couple of examples.

**Example Evaluate the double
integral**

$$I = \int_1^2 \int_1^2 \frac{dx dy}{x + y}$$

**by using trapezoidal rule, with
h = k = 0.25.**

Solution Taking $x = 1, 1.25, 1.50, 1.75, 2.0$ and $y = 1, 1.25, 1.50, 1.75, 2.0$, the following table is generated using the integrand

$$f(x, y) = \frac{1}{x + y}$$

x	y				
	1.00	1.25	1.50	1.75	2.00
1.00	0.5	0.4444	0.4	0.3636	0.3333
1.25	0.4444	0.4	0.3636	0.3333	0.3077
1.50	0.4	0.3636	0.3333	0.3077	0.2857
1.75	0.3636	0.3333	0.307	0.2857	0.2667
2.00	0.3333	0.3077	0.2857	0.2667	0.25

Keeping one variable say x fixed and varying the variable y , the application of trapezoidal rule to each row in the above table gives

$$\int_1^2 f(1, y) dy$$

$$= \frac{0.25}{2} [0.5 + 2(0.4444 + 0.4$$
$$+ 0.3636) + 0.3333]$$

$$= 0.4062$$

$$\int_1^2 f(1.25, y) dy$$

$$\begin{aligned} &= \frac{0.25}{2} [0.4444 + 2(0.4 + \\ &\quad 0.3636 + 0.3333) + 0.3077] \\ &= 0.3682 \end{aligned}$$

$$\int_1^2 f(1.5, y) dy$$

$$= \frac{0.25}{2} [0.4 + 2(0.3636 + 0.3333 + 0.3077)] + 0.2857$$
$$= 0.3369$$

$$\int_1^2 f(1.75, y) dy$$

$$= \frac{0.25}{2} [0.3636 + 2(0.3333 +$$

$$0.3077 + 0.2857) + 0.2667]$$

$$= 0.3105$$

and

$$\int_1^2 f(2, y) dy$$

$$= \frac{0.25}{2} [0.3333 + 2(0.3077$$

$$+ 0.2857) + 0.25]$$

$$= 0.2879$$

Therefore,

$$\begin{aligned} I &= \int_1^2 \int_1^2 \frac{dx dy}{x + y} \\ &= \frac{h}{2} \{ f(1, y) + 2[f(1.25, y) + \\ &f(1.5, y) + f(1.75, y)] + f(2, y) \} \end{aligned}$$

**By use of the last equations
we get the required result as**

$$I = \frac{0.25}{2} \left[.04602 + 2(0.3682 + 0.3369 + 0.3105) + 0.2879 \right] = 0.3407$$

Example :Evaluate

$$\int_0^{\pi/2} \int_0^{\pi/2} \sqrt{\sin(x+y)} dx dy$$

**by numerical double
integration.**

Solution Taking $x = y = \pi/4, 3\pi/8, \pi/2$, we can generate the following table of the integrand

$$f(x, y) = \sqrt{\sin(x + y)}$$

x	y				
	0	$\pi/8$	$\pi/4$	$3\pi/8$	$\pi/2$
0	0.0	0.6186	0.8409	0.9612	1.0
$\pi/8$	0.6186	0.8409	0.9612	1.0	0.9612
$\pi/4$	0.8409	0.9612	1.0	0.9612	0.8409
$3\pi/8$	0.9612	1.0	0.9612	0.8409	0.6186
$\pi/2$	1.0	0.9612	0.8409	0.618	0.0

Keeping one variable as say x fixed and y as variable, and applying trapezoidal rule to each row of the above table, we get

$$\int_0^{\pi/2} f(0, y) dx$$

$$= \frac{\pi}{16} [0.0 + 2(0.6186 + 0.8409 + 0.9612) + 1.0] = 1.1469$$

$$\int_0^{\pi/2} \left(\frac{\pi}{8}, y \right) dx$$

$$= \frac{\pi}{16} [0.6186 + 2(0.8409 + 0.9612 + 1.0) + 0.9612] = 1.4106$$

Similarly, we get

$$\int_0^{\pi/2} f\left(\frac{\pi}{4}, y\right) dx = 1.4778,$$

$$\int_0^{\pi/2} f\left(\frac{3\pi}{8}, y\right) dx = 1.4106.$$

and

$$\int_0^{\pi/2} f\left(\frac{\pi}{2}, y\right) dx = 1.1469$$

Using these results, we finally obtain

$$\begin{aligned} & \int_0^{\pi/2} \int_0^{\pi/2} \sqrt{\sin(x+y)} dx dy \\ &= \frac{\pi}{16} \left\{ f(0, y) + 2 \left(\frac{\pi}{8}, y \right) + f \left(\frac{\pi}{4}, y \right) \right. \\ & \quad \left. + f \left(\frac{3\pi}{8}, y \right) + f \left(\frac{\pi}{2}, y \right) \right\} \end{aligned}$$

$$\begin{aligned} &= \frac{\pi}{16} [1.1469 + 2(1.4106 \\ &\quad + 1.4778 + 1.4106) + 1.1469] \\ &= 2.1386 \end{aligned}$$

Numerical Analysis

Lecture 34

Numerical Analysis Lecture 35

Chapter 7

Ordinary Differential Equations

Introduction

Taylor Series

Euler Method

Runge-Kutta Method

Predictor Corrector

Method

INTRODUCTION

Many problems in science and engineering when formulated mathematically are readily expressed in terms of ordinary differential equations (ODE) with initial and boundary condition.

Example

The trajectory of a ballistic missile, the motion of an artificial satellite in its orbit, are governed by ordinary differential equations.

Theories concerning electrical networks, bending of beams, stability of aircraft, etc., are modeled by differential equations.

To be more precise, the rate of change of any quantity with respect to another can be modeled by an ODE

**Closed form solutions
may not be possible to
obtain, for every
modeled problem, while
numerical methods
exist, to solve them
using computers.**

In general, a linear or non-linear ordinary differential equation can be written as

$$\frac{d^n y}{dt^n} = f\left(t, y, \frac{dy}{dt}, \dots, \frac{d^{n-1} y}{dt^{n-1}}\right)$$

**Here we shall focus on a
system of first order
differential equations of the
form**

$$\frac{dy}{dt} = f(t, y)$$

**with the initial condition
 $y(t_0) = y_0$, which is called an
initial value problem (IVP).**

It is justified, in view of the fact that any higher order ODE can be reduced to a system of first order differential equations by substitution.

**For example, consider
a second order
differential equation
of the form**

$$y'' = f(t, y, y')$$

Introducing the substitution $p = y'$, the above equation reduces to a system of two first order differential equations, such as

$$y' = p, \quad p' = f(t, y, p)$$

Theorem

Let $f(t, y)$ be real and continuous in the strip R , defined by

$$t \in [t_0, T], \quad -\infty \leq y \leq \infty$$

Then for any $t \in [t_0, T]$ and for any y_1, y_2 , there exists a constant L , satisfying the inequality $|f(t, y_1) - f(t, y_2)| \leq L|y_1 - y_2|$

so that $|f_y(t, y)| \leq L,$

for every $t, y \in R.$

Here, L is called *Lipschitz constant*.

If the above conditions are satisfied, then for any y_0 , the IVP has a unique solution $y(t)$, for $t \in [t_0, T]$

In fact, we assume the existence and uniqueness of the solution to the above IVP

The function may be linear or non-linear. We also assume that the function $f(t, y)$ is sufficiently differentiable with respect to either t or y .

TAYLOR'S SERIES METHOD

Consider an initial value problem described by

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0$$

Here, we assume that $f(t, y)$ is sufficiently differentiable with respect to x and y .

**If $y(t)$ is the exact solution,
we can expand $y(t)$ by
Taylor's series about the
point $t = t_0$ and obtain**

$$y(t) = y(t_0) + (t - t_0)y'(t_0) + \frac{(t - t_0)^2}{2!}y''(t_0)$$

$$+ \frac{(t - t_0)^3}{3!}y'''(t_0) + \frac{(t - t_0)^4}{4!}y^{IV}(t_0) + \dots$$

Since, the solution is not known, the derivatives in the above expansion are not known explicitly. However, f is assumed to be sufficiently differentiable and therefore, the derivatives can be obtained directly from the given differential equation.

Noting that f is an implicit function of y , we have

$$\left. \begin{aligned} y' &= f(t, y) \\ y'' &= \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \frac{dy}{dx} = f_x + f f_y \end{aligned} \right\}$$

Similarly

$$\left. \begin{aligned} y''' &= f_{xx} + ff_{xy} + f(f_{xy} + ff_{yy}) + f_y(f_x + ff_y) \\ &= f_{xx} + 2ff_{xy} + f^2 f_{yy} + f_y(f_x + ff_y) \\ y^{IV} &= f_{xxx} + 3ff_{xxx} + 3f^2 f_{xyy} \\ &\quad + f_y(f_{xx} + 2ff_{xy} + f^2 f_{yy}) \\ &\quad + 3(f_x + ff_y)(ff_{xy} + ff_{yy}) \\ &\quad + f_y^2(f_x + ff_y) \end{aligned} \right\}$$

**Continuing in this manner,
we can express any
derivative of y in terms of
 $f(t, y)$ and its partial
derivatives.**

Example

Using Taylor's series method,
find the solution of the initial
value problem

$$\frac{dy}{dt} = t + y, \quad y(1) = 0$$

at $t = 1.2$, with $h = 0.1$ and
compare the result with the
closed form solution

Solution_

Let us compute the first few derivatives from the given differential equation as follows:

$$\begin{aligned}y' &= t + y, & y'' &= 1 + y', & y''' &= y'', \\y^{IV} &= y''', & y^V &= y^{IV}\end{aligned}$$

Prescribing the initial condition, that is, at $t_0 = 1$, $y_0 - y(t_0) = 0$, we have

$$y_0' = 1, y_0'' = 2,$$

$$y_0''' = y_0^{IV} = y_0^V = 2$$

Now, using Taylor's series method, we have

$$y(t) = y_0 + (t - t_0)y'_0 + \frac{(t - t_0)^2}{2}y''_0 + \frac{(t - t_0)^3}{6}y'''_0 \\ + \frac{(t - t_0)^4}{24}y^{IV}_0 + \frac{(t - t_0)^5}{120}y^V_0 + \dots$$

Substituting the above values of the derivatives, and the initial condition, we obtain

$$y(1.1) = 0 + (0.1)(1) + \frac{0.01}{2}(2) + \frac{0.001}{6}(2) + \frac{0.00001}{120}(2) + \dots$$

$$= 0.1 + 0.01 + \frac{0.001}{3} + \frac{0.0001}{12} + \frac{0.00001}{60} + \dots$$

$$= 0.1 + 0.01 + 0.000333 + 0.0000083 + 0.0000001 + \dots$$

$$= 0.1103414.$$

Therefore,

$$y(1.1) = y_1 = 0.1103414 \cong 0.1103$$

**Taking $y_1 = 0.1103$ at $t = 1.1$,
the values of the derivatives
are**

$$y_1' = 1.1 + 0.1103 = 1.2103$$

$$y_1'' = 1 + 1.2103 = 2.2103$$

$$y_1''' = y_1^{IV} = y_1^V = 2.2103$$

Substituting the value of y_1 and its derivatives into Taylor's series expansion we get, after retaining terms up to fifth derivative only.....

$$y(1.2) = y_1 + (t - t_1)y_1' + \frac{(t - t_1)^2}{2}y_1'' + \frac{(t - t_1)^3}{6}y_1''' + \frac{(t - t_1)^4}{24}y_1^{IV} + \frac{(t - t_1)^5}{120}y_1^V$$

$$y(1.2) = 0.1103 + 0.12103 \\ + 0.0110515 + 0.0003683 \\ + 0.000184 + 0.00000003$$

$$= 0.2429341 \square 0.2429 \approx 0.243$$

To obtain the closed form solution, we rewrite the given IVP as

$$\frac{dy}{dt} - y = t \quad \text{or} \quad d(ye^{-t}) = te^{-t}$$

On integration, we get

$$\begin{aligned} y &= -e^t (te^{-t} + e^{-t}) + ce^t \\ &= ce^t - t - 1 \end{aligned}$$

Using the initial condition, we

get $0 = ce - 2$ or $c = \frac{2}{e}$

Therefore, the closed form solution is

$$y = -t - 1 + 2e^{t-1}$$

**When $t = 1.2$,
the closed form solution
becomes**

$$\begin{aligned}y(1.2) &= -1.2 - 1 + 2(1.2214028) \\&\quad - 2.2 + 2.4428056 \\&= 0.2428 \approx 0.243\end{aligned}$$

Numerical Analysis Lecture 35

Numerical Analysis Lecture 36

Chapter 7

Ordinary Differential Equations

Introduction

Taylor Series

Euler Method

Runge-Kutta Method

Predictor Corrector

Method

TAYLOR'S SERIES METHOD

**We considered an
initial value problem
described by**

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0$$

We expanded $y(t)$ by Taylor's series about the point $t = t_0$ and obtain

$$y(t) = y(t_0) + (t - t_0)y'(t_0) + \frac{(t - t_0)^2}{2!}y''(t_0)$$

$$+ \frac{(t - t_0)^3}{3!}y'''(t_0) + \frac{(t - t_0)^4}{4!}y^{IV}(t_0) + \dots$$

Noting that f is an implicit function of y , we have

$$\left. \begin{aligned} y' &= f(t, y) \\ y'' &= \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \frac{dy}{dx} = f_x + ff_y \end{aligned} \right\}$$

Similarly

$$\left. \begin{aligned} y''' &= f_{xx} + ff_{xy} + f(f_{xy} + ff_{yy}) + f_y(f_x + ff_y) \\ &= f_{xx} + 2ff_{xy} + f^2 f_{yy} + f_y(f_x + ff_y) \\ y^{IV} &= f_{xxx} + 3ff_{xxx} + 3f^2 f_{xyy} \\ &\quad + f_y(f_{xx} + 2ff_{xy} + f^2 f_{yy}) \\ &\quad + 3(f_x + ff_y)(ff_{xy} + ff_{yy}) \\ &\quad + f_y^2(f_x + ff_y) \end{aligned} \right\}$$

EULER METHOD

Euler method is one of the oldest numerical methods used for integrating the ordinary differential equations. Though this method is not used in practice, its understanding will help us to gain insight into nature of predictor-corrector method

Consider the differential equation of first order with the initial condition $y(t_0) = y_0$.

$$\frac{dy}{dt} = f(t, y)$$

The integral of this equation is a curve in, ty -plane.

Here, we find successively $y_1, y_2, \dots, y_m$; where y_m is the value of y at $t = t_m = t_0 + mh, m = 1, 2, \dots$ and h being small.

Here we use a property that in a small interval, a curve is nearly a straight line. Thus at (t_0, y_0) , we approximate the curve by a tangent at that point.

Therefore,
$$\left(\frac{dy}{dt}\right)_{(t_0, y_0)} = \frac{y - y_0}{t - t_0} = f(t_0, y_0)$$

That is,
$$y = y_0 + (t - t_0)f(t_0, y_0)$$

Hence, the value of y corresponding to $t = t_1$ is given by

$$y_1 = y_0 + (t_1 - t_0)f(t_0, y_0)$$

Similarly approximating the solution curve in the next interval (t_1, t_2) by a line through (t_1, y_1) having its slope $f(t_1, y_1)$, we obtain

$$y_2 = y_1 + hf(t_1, y_1)$$

Thus, we obtain in general, the solution of the given differential equation in the form of a recurrence relation

$$y_{m+1} = y_m + hf(t_m, y_m)$$

Geometrically, this method has a very simple meaning. The desired function curve is approximated by a polygon train, where the direction of each part is determined by the value of the function $f(t, y)$ at its starting point.

Example

Given
$$\frac{dy}{dt} = \frac{y - t}{y + t}$$

with the initial condition $y = 1$ at $t = 0$. Using Euler method, find y approximately at $x = 0.1$, in five steps.

Solution

Since the number of steps are five, we shall proceed in steps of $(0.1)/5 = 0.02$.

Therefore, taking step size $h = 0.02$, we shall compute the value of y at $t = 0.02, 0.04, 0.06, 0.08$ and 0.1

Thus $y_1 = y_0 + hf(t_0, y_0)$, **where** $y_0 = 1, t_0 = 0$

Therefore, $y_1 = 1 + 0.02 \frac{1-0}{1+0} = 1.02$

Similarly,

$$y_2 = y_1 + hf(t_1, y_1) = 1.02 + 0.02 \frac{1.02 - 0.02}{1.02 + 0.02} = 1.0392$$

$$y_3 = y_2 + hf(t_2, y_2) = 1.0392 + 0.02 \frac{1.0392 - 0.04}{1.0392 + 0.04} = 1.0577$$

$$y_4 = y_3 + hf(t_3, y_3) = 1.0577 + 0.02 \frac{1.0577 - 0.06}{1.0577 + 0.06} = 1.0738$$

$$y_5 = y_4 + hf(t_4, y_4) = 1.0738 + 0.02 \frac{1.0738 - 0.08}{1.0738 + 0.08} = 1.0910$$

**Hence the value of
 y corresponding to
 $t = 0.1$ is 1.091**

MODIFIED EULER'S METHOD

The modified Euler's method gives greater improvement in accuracy over the original Euler's method. Here, the core idea is that we use a line through (t_0, y_0) whose slope is the average of the slopes at (t_0, y_0) and $(t_1, y_1^{(1)})$

**Where $y_1^{(1)} = y_0 + hf(t_0, y_0)$
is the value of y at $t = t_1$ as
obtained in Euler's method,
which approximates the curve
in the interval (t_0, t_1)**

Figure

Geometrically, from Figure, if L_1 is the tangent at (t_0, y_0) , L_2 is the line through $(t_1, y_1^{(1)})$ of slope $f(t_1, y_1^{(1)})$ and L^- is the line through $(t_1, y_1^{(1)})$ but with a slope equal to the average of $f(t_0, y_0)$ and $f(t_1, y_1^{(1)})$,.....

the line L through (t_0, y_0) and parallel to $\bar{L}$ is used to approximate the curve in the interval (t_0, t_1) .

Thus, the ordinate of the point B will give the value of y_1 .

**Now, the equation of the line
AL is given by**

$$y_1 = y_0 + \left[\frac{f(t_0, y_0) + f(t_1, y_1^{(1)})}{2} \right] (t_1 - t_0)$$

$$= y_0 + h \left[\frac{f(t_0, y_0) + f(t_1, y_1^{(1)})}{2} \right]$$

Similarly proceeding, we arrive at the recurrence relation

$$y_{m+1} = y_m + h \left[\frac{f(t_m, y_m) + f(t_{m+1}, y_{m+1}^{(1)})}{2} \right]$$

This is the *modified Euler's method*.

Example

Using modified Euler's method, obtain the solution of the differential equation

$$\frac{dy}{dt} = t + \sqrt{y} = f(t, y)$$

with the initial condition

$y_0 = 1$ at $t_0 = 0$ for the range

$0 \leq t \leq 0.6$ in steps of 0.2

Solution

At first, we use Euler's method to get

$$\begin{aligned} y_1^{(1)} &= y_0 + hf(t_0, y_0) \\ &= 1 + 0.2(0 + 1) = 1.2 \end{aligned}$$

Then, we use modified Euler's method to find

$$y(0.2) = y_1 = y_0 + h \frac{f(t_0, y_0) + f(t_1, y_1^{(1)})}{2}$$

$$= 1.0 + 0.2 \frac{1 + (0.2 + \sqrt{1.2})}{2} = 1.2295$$

Similarly proceeding, we have from Euler's method

$$\begin{aligned}y_2^{(1)} &= y_1 + hf(t_1, y_1) \\&= 1.2295 + 0.2(0.2 + \sqrt{1.2295}) \\&= 1.4913\end{aligned}$$

**Using modified Euler's method,
we get**

$$y_2 = y_1 + h \frac{f(t_1, y_1) + f(t_2, y_2^{(1)})}{2}$$

$$= 1.2295 + 0.2 \frac{\left(0.2 + \sqrt{1.2295}\right) + \left(0.4 + \sqrt{1.4913}\right)}{2}$$

$$= 1.5225$$

Finally

$$\begin{aligned}y_3^{(1)} &= y_2 + hf(t_2, y_2) \\&= 1.52225 + 0.2 \left(0.4 + \sqrt{1.52225} \right) \\&= 1.8493\end{aligned}$$

Modified Euler's method gives

$$y(0.6) = y_3 = y_2 + h \frac{f(t_2, y_2) + f(t_3, y_3^{(1)})}{2}$$

$$= 1.5225 + 0.1 \left[(0.4 + \sqrt{1.5225}) + (0.6 + \sqrt{1.8493}) \right]$$

$$= 1.8819$$

Hence, the solution to the given problem is given by

t	0.2	0.4	0.6
y	1.2295	1.5225	1.8819

Numerical Analysis Lecture 36

Numerical Analysis Lecture 37

Chapter 7

Ordinary Differential Equations

Introduction

Taylor Series

Euler Method

Runge-Kutta Method

Predictor Corrector

Method

TAYLOR'S SERIES METHOD

We considered an initial value problem described by

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0$$

We expanded $y(t)$ by Taylor's series about the point $t = t_0$ and obtain

$$y(t) = y(t_0) + (t - t_0)y'(t_0) + \frac{(t - t_0)^2}{2!}y''(t_0)$$

$$+ \frac{(t - t_0)^3}{3!}y'''(t_0) + \frac{(t - t_0)^4}{4!}y^{IV}(t_0) + \dots$$

Noting that f is an implicit function of y , we have

$$\left. \begin{aligned} y' &= f(t, y) \\ y'' &= \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \frac{dy}{dx} = f_x + f f_y \end{aligned} \right\}$$

Similarly

$$\left. \begin{aligned} y''' &= f_{xx} + ff_{xy} + f(f_{xy} + ff_{yy}) + f_y(f_x + ff_y) \\ &= f_{xx} + 2ff_{xy} + f^2 f_{yy} + f_y(f_x + ff_y) \\ y^{IV} &= f_{xxx} + 3ff_{xxx} + 3f^2 f_{xyy} \\ &\quad + f_y(f_{xx} + 2ff_{xy} + f^2 f_{yy}) \\ &\quad + 3(f_x + ff_y)(ff_{xy} + ff_{yy}) \\ &\quad + f_y^2(f_x + ff_y) \end{aligned} \right\}$$

EULER METHOD

Consider the differential equation of first order with the initial condition $y(t_0) = y_0$.

$$\frac{dy}{dt} = f(t, y)$$

**The value of y corresponding
to $t = t_1$**

$$y_1 = y_0 + (t_1 - t_0)f(t_0, y_0)$$

Similarly

$$y_2 = y_1 + hf(t_1, y_1)$$

Then, we obtained the solution in the form of a recurrence relation

$$y_{m+1} = y_m + hf(t_m, y_m)$$

MODIFIED EULER'S METHOD

The recurrence relation

$$y_{m+1} = y_m + h \left[\frac{f(t_m, y_m) + f(t_{m+1}, y_{m+1}^{(1)})}{2} \right]$$

is the *modified Euler's method*.

RUNGE – KUTTA METHOD

**These are computationally,
most efficient methods in
terms of accuracy. They
were developed by two
German mathematicians,
Runge and Kutta.**

They are distinguished by their orders in the sense that they agree with Taylor's series solution up to terms of h^r , where r is the order of the method. These methods do not demand prior computation of higher derivatives of $y(t)$ as in TSM.

Fourth-order Runge-Kutta methods are widely used for finding the numerical solutions of linear or non-linear ordinary differential equations, the development of which is complicated algebraically.

Therefore, we convey the basic idea of these methods by developing the second-order Runge-Kutta method which we shall refer hereafter as R-K method.

Please recall that the modified Euler's method: which can be viewed as

$$y_{n+1} = y_n + h \text{ (average of slopes)}$$

This, in fact, is the basic idea of R-K method.

Here, we find the slopes not only at t_n but also at several other interior points, and take the weighted average of these slopes and add to y_n to get y_{n+1} . Now, we shall derive the second order R-K method in the following slides.

Consider the IVP

$$\frac{dy}{dt} = f(t, y), \quad y(t_n) = y_n$$

We also define

$$k_1 = hf(t_n, y_n), \quad k_2 = hf(t_n + \alpha h, y_n + \beta k_1)$$

**and take the weighted
average of k_1 and k_2 and add
to y_n to get y_{n+1}**

We seek a formula of the form

$$y_{n+1} = y_n + W_1 k_1 + W_2 k_2$$

Where α, β, W_1 and W_2 are constants to be determined so that the above equation agree with the Taylor's series expansion as high an order as possible

Thus, using Taylor's series expansion, we have

$$y(t_{n+1}) = y(t_n) + hy'(t_n) + \frac{h^2}{2} y''(t_n) + \frac{h^3}{6} y'''(t_n) + \dots$$

Rewriting the derivatives of y in terms of f of the above equation, we get

$$y_{n+1} = y_n + hf(t_n, y_n) + \frac{h^2}{2}(f_t + ff_y)$$

$$+ \frac{h^3}{6}[f_{tt} + 2ff_{ty} + f^2 f_{yy} + f_y(f_t + ff_y)] + O(h^4)$$

Here, all derivatives are evaluated at (t_n, y_n) .

Next, we shall rewrite the given equation after inserting the expressions for k_1 and k_2 as

$$y_{n+1} = y_n + W_1 h f(t_n, y_n) + W_2 h f(t_n + \alpha h, y_n + \beta k_1)$$

Now using Taylor's series expansion of two variables, we obtain

$$y_{n+1} = y_n + W_1 h f(t_n, y_n) + W_2 h \left[f(t_n, y_n) + (\alpha h f_t + \beta k_1 f_y) \right]$$

$$+ \left(\frac{\alpha^2 h^2}{2} f_{nn} + \alpha h \beta k_1 f_{ty} + \frac{\beta^2 k_1^2}{2} f_{yy} \right) + O(h^3) \Bigg]$$

Here again, all derivatives are computed at (t_n, y_n) . On inserting the expression for k_1 , the above equation becomes

$$y_{n+1} = y_n + (W_1 + W_2)hf + W_2h \left[(\alpha hf_t + \beta hff_y) \right.$$

$$\left. + \left(\frac{\alpha^2 h^2}{2} f_{tt} + \alpha\beta h^2 ff_{ty} + \frac{\beta^2 h^2}{2} f^2 f_{yy} \right) + O(h^3) \right]$$

On rearranging in the increasing powers of h , we get

$$y_{n+1} = y_n + (W_1 + W_2)hf + W_2h^2(\alpha f_t + \beta ff_y)$$

$$+ W_2h^3 \left(\frac{\alpha^2}{2} f_{tt} + \alpha\beta ff_{ty} + \frac{\beta^2 f^2 ff_y}{2} \right) + O(h^4) \Bigg]$$

Now, equating coefficients of h and h^2 in the two equations, we obtain

$$W_1 + W_2 = 1, \quad W_2(\alpha f_t + \beta ff_y) = \frac{f_t + ff_y}{2}$$

Implying

$$W_1 + W_2 = 1, \quad W_2\alpha = W_2\beta = \frac{1}{2}$$

Thus, we have three equations in four unknowns and so, we can choose one value arbitrarily. Solving we get

$$W_1 = 1 - W_2, \quad \alpha = \frac{1}{2W_2}, \quad \beta = \frac{1}{2W_2}$$

where W_2 is arbitrary and various values can be assigned to it

**We now consider two cases,
which are popular**

Case I

**If we choose $W_2 = 1/3$,
then $W_1 = 2/3$ and $\alpha = \beta = 3/2$.**

$$y_{n+1} = y_n + \frac{1}{3}(2k_1 + k_2)$$

$$k_1 = hf(t, y), \quad k_2 = hf\left(t + \frac{3}{2}h, y + \frac{3}{2}k_1\right)$$

Case II: If we consider $W_2 = 1/2$, then $W_1 = 1/2$ and

$$\alpha = \beta = 1.$$

Then

$$y_{n+1} = y_n + \frac{k_1 + k_2}{2}$$

$$k_1 = hf(t, y), \quad k_2 = hf(t + h, y + k_1)$$

In fact, we can recognize that this equation is the modified Euler's method and is therefore a special case of a 2nd order Runge-Kutta method. These equations are known as 2nd order R –K Methods, since they agree with Taylor's series solution up to the term h^2 .

Defining the local truncation error, TE, as the difference between the exact solution $y(t_{n+1})$ at $t = t_{n+1}$ and the numerical solution y_{n+1} , obtained using the second order R – K method, we have

$$TE = y(t_{n+1}) - y_{n+1}$$

Now, substituting

$$W_2 = \frac{1}{2\alpha}, \quad W_1 = 1 - \frac{1}{2\alpha}, \quad \beta = \alpha,$$

into the above equation, we get

$$y_{n+1} = y_n + hf_{tt} + \frac{h^2}{2}(f_t + ff_y)_{t=t_n}$$

$$+ \frac{h^3\alpha}{4}(f_{tt} + 2ff_{ty} + f^2 f_{yy})_{t=t_n} + \dots$$

Finally, we obtain

$$TE = h^3 \left[\left(\frac{1}{6} - \frac{\alpha}{4} \right) (f_{tt} + 2ff_{ty} + f^2 f_{yy}) + \frac{1}{6} f_y (f_t + ff_y) \right]$$

The expression can further be simplified to

$$TE = h^3 \left[\left(\frac{1}{6} - \frac{\alpha}{4} \right) (y''' - f_y y') + \frac{1}{6} f_y y' \right]$$

Therefore, the expression for local truncation error is given by

$$TE = h^3 \left[\left(\frac{1}{6} - \frac{\alpha}{4} \right) y''' + \frac{\alpha}{4} f_y y' \right]$$

Please verify that the magnitude of the TE in case I is less than that of case II

Following similar procedure, Runge-Kutta formulae of any order can be obtained.

However, their derivations becomes exceedingly lengthy and complicated.

Amongst them, the most popular and commonly used in practice is the R-K method of fourth-order, which agrees with the Taylor series method up to terms of $O(h^4)$.

This well-known fourth-order R-K method is described in the following steps.

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

where

$$k_1 = hf(t_n, y_n)$$

$$k_2 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$

$$k_3 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right)$$

$$k_4 = hf(t_n + h, y_n + k_3)$$

Please note that the second-order Runge-Kutta method described above requires the evaluation of the function twice for each complete step of integration.

Similarly, fourth-order Runge-Kutta method requires the evaluation of the function four times. The discussion on optimal order R-K method is very interesting, but will be discussed some other time.

Numerical Analysis Lecture 37

Numerical

Analysis

Lecture 38

Chapter 7

Ordinary Differential Equations

Introduction

Taylor Series

Euler Method

Runge-Kutta Method

Predictor Corrector

Method

TAYLOR'S SERIES METHOD

We considered an initial value problem described by

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0$$

We expanded $y(t)$ by Taylor's series about the point $t = t_0$ and obtain

$$y(t) = y(t_0) + (t - t_0)y'(t_0) + \frac{(t - t_0)^2}{2!}y''(t_0)$$

$$+ \frac{(t - t_0)^3}{3!}y'''(t_0) + \frac{(t - t_0)^4}{4!}y^{IV}(t_0) + \dots$$

Noting that f is an implicit function of y , we have

$$\left. \begin{aligned} y' &= f(t, y) \\ y'' &= \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \frac{dy}{dx} = f_x + ff_y \end{aligned} \right\}$$

Similarly

$$\left. \begin{aligned} y''' &= f_{xx} + ff_{xy} + f(f_{xy} + ff_{yy}) + f_y(f_x + ff_y) \\ &= f_{xx} + 2ff_{xy} + f^2 f_{yy} + f_y(f_x + ff_y) \\ y^{IV} &= f_{xxx} + 3ff_{xxx} + 3f^2 f_{xyy} \\ &\quad + f_y(f_{xx} + 2ff_{xy} + f^2 f_{yy}) \\ &\quad + 3(f_x + ff_y)(ff_{xy} + ff_{yy}) \\ &\quad + f_y^2(f_x + ff_y) \end{aligned} \right\}$$

EULER METHOD

Consider the differential equation of first order with the initial condition $y(t_0) = y_0$.

$$\frac{dy}{dt} = f(t, y)$$

**The value of y corresponding
to $t = t_1$**

$$y_1 = y_0 + (t_1 - t_0)f(t_0, y_0)$$

Similarly

$$y_2 = y_1 + hf(t_1, y_1)$$

**Then, we obtained the
solution in the form of a
recurrence relation**

$$y_{m+1} = y_m + hf(t_m, y_m)$$

MODIFIED EULER'S METHOD

The recurrence relation

$$y_{m+1} = y_m + h \left[\frac{f(t_m, y_m) + f(t_{m+1}, y_{m+1}^{(1)})}{2} \right]$$

is the *modified Euler's method*.

RUNGE – KUTTA METHODS

We considered the IVP

$$\frac{dy}{dt} = f(t, y), \quad y(t_n) = y_n$$

We also defined

$$k_1 = hf(t_n, y_n), \quad k_2 = hf(t_n + \alpha h, y_n + \beta k_1)$$

**and took the weighted
average of k_1 and k_2 and
added to y_n to get y_{n+1}**

We obtained

$$W_1 + W_2 = 1, \quad W_2(\alpha f_t + \beta ff_y) = \frac{f_t + ff_y}{2}$$

Implying

$$W_1 + W_2 = 1, \quad W_2\alpha = W_2\beta = \frac{1}{2}$$

We considered two cases,

**Case I We choose $W_2 = 1/3$,
then $W_1 = 2/3$ and $\alpha = \beta = 3/2$.**

$$y_{n+1} = y_n + \frac{1}{3}(2k_1 + k_2)$$

$$k_1 = hf(t, y), \quad k_2 = hf\left(t + \frac{3}{2}h, y + \frac{3}{2}k_1\right)$$

Case II: We considered
 $W_2 = 1/2$, then $W_1 = 1/2$ and

$$\alpha = \beta = 1.$$

Then

$$y_{n+1} = y_n + \frac{k_1 + k_2}{2}$$

$$k_1 = hf(t, y), \quad k_2 = hf(t + h, y + k_1)$$

Finally, we obtain

$$TE = h^3 \left[\left(\frac{1}{6} - \frac{\alpha}{4} \right) (f_{tt} + 2ff_{ty} + f^2 f_{yy}) + \frac{1}{6} f_y (f_t + ff_y) \right]$$

The expression can further be simplified to

$$TE = h^3 \left[\left(\frac{1}{6} - \frac{\alpha}{4} \right) (y''' - f_y y') + \frac{1}{6} f_y y' \right]$$

Therefore, the expression for local truncation error is given by

$$TE = h^3 \left[\left(\frac{1}{6} - \frac{\alpha}{4} \right) y''' + \frac{\alpha}{4} f_y y' \right]$$

The fourth-order R-K method was described as

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

where

$$k_1 = hf(t_n, y_n)$$

$$k_2 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$

$$k_3 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right)$$

$$k_4 = hf(t_n + h, y_n + k_3)$$

Example

Use the following second order Runge-Kutta method described by

$$y_{n+1} = y_n + \frac{1}{3}(2k_1 + k_2)$$

where

$$k_1 = hf(x_n, y_n) \quad \text{and} \quad k_2 = hf\left(x_n + \frac{3}{2}h, y_n + \frac{3}{2}k_1\right)$$

and find the numerical solution of the initial value problem described as

$$\frac{dy}{dx} = \frac{y+x}{y-x}, \quad y(0) = 1$$

at $x = 0.4$ and taking $h = 0.2$.

Solution

Here $f(x, y) = \frac{y+x}{y-x}, \quad h = 0.2, \quad x_0 = 0, \quad y_0 = 1$

We calculate

$$k_1 = hf(x_0, y_0) = 0.2 \frac{1+0}{1-0} = 0.2$$

$$\begin{aligned} k_2 &= hf[x_0 + 0.3, y_0 + (1.5)(0.2)] \\ &= hf(0.3, 1.3) = 0.2 \frac{1.3+0.3}{1.3-0.3} = 0.32 \end{aligned}$$

Now, using the given R-K method, we get

$$y(0.2) = y_1 = 1 + \frac{1}{3}(0.4 + 0.32) = 1.24$$

**Now, taking $x_1 = 0.2$,
 $y_1 = 1.24$, we calculate**

$$k_1 = hf(x_1, y_1) = 0.2 \frac{1.24 + 0.2}{1.24 - 0.2} = 0.2769$$

$$k_2 = hf \left(x_1 + \frac{3}{2}h, y_1 + \frac{3}{2}k_1 \right) = hf(0.5, 1.6554) \\ = 0.2 \frac{1.6554 + 0.5}{1.6554 - 0.5} = 0.3731$$

Again using the given R-K method, we obtain

$$y(0.4) = y_2 = 1.24 + \frac{1}{3} [2(0.2769) + 0.3731] \\ = 1.54897$$

Example

Solve the following differential equation

$$\frac{dy}{dt} = t + y$$

with the initial condition $y(0) = 1$,
using fourth- order Runge-Kutta
method from $t = 0$ to $t = 0.4$
taking $h = 0.1$

Solution

The fourth-order Runge-Kutta method is described as

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (1)$$

where

$$k_1 = hf(t_n, y_n)$$

$$k_2 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$

$$k_3 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right)$$

$$k_4 = hf(t_n + h, y_n + k_3)$$

In this problem,

$$f(t, y) = t + y, \quad h = 0.1, \quad t_0 = 0, \quad y_0 = 1.$$

As a first step, we calculate

$$k_1 = hf(t_0, y_0) = 0.1(1) = 0.1$$

$$\begin{aligned} k_2 &= hf(t_0 + 0.05, y_0 + 0.05) \\ &= hf(0.05, 1.05) = 0.1[0.05 + 1.05] \\ &= 0.11 \end{aligned}$$

$$\begin{aligned} k_3 &= hf(t_0 + 0.05, y_0 + 0.055) \\ &= 0.1(0.05 + 1.055) \\ &= 0.1105 \end{aligned}$$

$$k_4 = 0.1(0.1 + 1.1105) = 0.12105$$

Now, we compute from

$$\begin{aligned}y_1 &= y_0 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\&= 1 + \frac{1}{6}(0.1 + 0.22 + 0.2210 + 0.12105) \\&= 1.11034\end{aligned}$$

Therefore $y(0.1) = y_1 = 1.1103$

In the second step, we have to find $y_2 = y(0.2)$

We compute

$$k_1 = hf(t_1, y_1) = 0.1(0.1 + 1.11034) = 0.121034$$

$$k_2 = hf\left(t_1 + \frac{h}{2}, y_1 + \frac{k_1}{2}\right)$$

$$= 0.1[0.15 + (1.11034 + 0.060517)] = 0.13208$$

$$k_3 = hf\left(t_1 + \frac{h}{2}, y_1 + \frac{k_2}{2}\right)$$

$$= 0.1[0.15 + (1.11034 + 0.06604)] = 0.132638$$

$$k_4 = hf(t_1 + h, y_1 + k_3)$$

$$= 0.1[0.2 + (1.11034 + 0.132638)] = 0.1442978$$

From Equation (1), we see that

$$y_2 = 1.11034 + \frac{1}{6}[0.121034 + 2(0.13208) + 2(0.132638) + 0.1442978] = 1.2428$$

Similarly we calculate,

$$k_1 = hf(t_2, y_2) = 0.1[0.2 + 1.2428] = 0.14428$$

$$k_2 = hf\left(t_2 + \frac{h}{2}, y_2 + \frac{k_1}{2}\right) = 0.1[0.25 + (1.2428 + 0.07214)] = 0.156494$$

$$k_3 = hf\left(t_1 + \frac{h}{2}, y_1 + \frac{k_2}{2}\right) = 0.1[0.3 + (1.2428 + 0.078247)] = 0.1571047$$

$$k_4 = hf(t_2 + h, y_2 + k_3) = 0.1[0.3 + (1.2428 + 0.1571047)] = 0.16999047$$

Using equation (1), we compute

$$y(0.3) = y_3 = y_2 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 1.399711$$

Finally, we calculate

$$k_1 = hf(t_3, y_3) = 0.1[0.3 + 1.3997] = 0.16997$$

$$k_2 = hf\left(t_3 + \frac{h}{2}, y_3 + \frac{k_1}{2}\right) = 0.1[0.35 + (1.3997 + 0.084985)] = 0.1834685$$

$$k_3 = hf\left(t_3 + \frac{h}{2}, y_3 + \frac{k_2}{2}\right) = 0.1[0.35 + (1.3997 + 0.091734)] = 0.1841434$$

$$k_4 = hf(t_3 + h, y_3 + k_3) = 0.1[0.4 + (1.3997 + 0.1841434)] = 0.19838434$$

**Using them in equation (1),
we get**

$$\begin{aligned}y(0.4) &= y_4 \\&= y_3 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\&= 1.58363\end{aligned}$$

which is the required result.

RUNGE – KUTTA METHOD FOR SOLVING A SYSTEM OF EQUATIONS

The fourth-order Runge-Kutta method can be extended to numerically solve the higher-order ordinary differential equations- linear or non-linear

For illustration, let us consider a second order ordinary differential equation of the form

$$\frac{d^2 y}{dt^2} = f\left(t, y, \frac{dy}{dt}\right)$$

Using the substitution $\frac{dy}{dt} = p$

this equation can be reduced to two first-order simultaneous differential equations, as given by

$$\frac{dy}{dt} = p = f_1(t, y, p), \quad \frac{dp}{dt} = f_2(t, y, p)$$

Now, we can directly write down the Runge-Kutta fourth-order formulae for solving the system.

Let the initial conditions of the above system be given by

$$y(x_n) = y_n, \quad y'(x_n) = p(x_n) = p_n$$

Then, we define

$$\left. \begin{aligned} k_1 &= hf_1(t_n, y_n, p_n), & l_1 &= hf_2(t_n, y_n, p_n) \\ k_2 &= hf_1\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}, p_n + \frac{l_1}{2}\right), & l_2 &= hf_2\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}, p_n + \frac{l_1}{2}\right) \\ k_3 &= hf_1\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}, p_n + \frac{l_2}{2}\right), & l_3 &= hf_2\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}, p_n + \frac{l_2}{2}\right) \\ k_4 &= hf_1(t_n + h, y_n + k_3, p_n + l_3), & l_4 &= hf_2(t_n + h, y_n + k_3, p_n + l_3) \end{aligned} \right\}$$

Now, using the initial conditions y_n, p_n and 4th -order R-K formula, we compute

$$\left. \begin{aligned} y_{n+1} &= y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ p_{n+1} &= p_n + \frac{1}{6}(l_1 + 2l_2 + 2l_3 + l_4) \end{aligned} \right\}$$

This method can be extended on similar lines to solve system of n first order differential equations.

Example

Solve the following equation

$$y'' - (0.1)(1 - y^2)y' + y = 0$$

Using 4th order Runge-Kutta method for $x = 0.2$, with the initial values $y(0) = 1$, $y'(0) = 0$

Solution

Let $\frac{dy}{dx} = p = f_1(x, y, p)$

Then $\frac{dp}{dx} = (0.1)(1 - y_2)p - y = f_2(x, y, p)$

Thus, the given equation reduced to two first-order equations.

**In the present problem, we
are given**

$$x_0 = 0, y_0 = 1, p_0 = y_0' = 0$$

Taking $h = 0.2$, we compute

$$k_1 = hf_1(x_0, y_0, p_0) = 0.2(0.0) = 0.0$$

$$l_1 = hf_2(x_0, y_0, p_0) = 0.2(0.0 - 1) = -0.2$$

$$k_2 = hf_1\left(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}, p_0 + \frac{l_1}{2}\right)$$

$$= hf_2(0.1, 1.0, -0.1) = -0.02$$

$$l_2 = hf_2\left(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}, p_0 + \frac{l_1}{2}\right)$$

$$= hf_2(0.1, 1.0, -0.1) = 0.2(-0.1) = -0.02$$

$$k_3 = hf_1\left(x_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}, p_0 + \frac{l_2}{2}\right)$$

$$= hf_1(0.1, 0.99, -0.1) = 0.2(-0.1) = -0.02$$

$$l_3 = hf_2\left(x_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}, p_0 + \frac{l_2}{2}\right) = hf_2(0.1, 0.99, -0.1)$$

$$= 0.2[(0.1)(0.0199)(-0.1) - 0.99] = -0.1980$$

$$k_4 = hf_1(x_0 + h, y_0 + k_3, p_0 + l_3)$$

$$= hf_1(0.2, 0.98, -0.1980) = -0.0396$$

$$l_4 = hf_2(x_0 + h, y_0 + k_3, p_0 + l_3) = hf_2(0.2, 0.98, -0.1980)$$

$$= 0.2[(0.1)(1 - 0.9604)(-0.1980) - 0.98] = -0.19616.$$

Now, $y(0.2) = y_1$ is given by

$$y(0.2) = y_1 = y_0 + \frac{1}{6}[k_1 + 2k_2 + 2k_3 + k_4]$$

$$= 1 + \frac{1}{6}[0.0 + 2(-0.02) + 2(-0.02) + (-0.0396)]$$

$$= 1 - 0.019935 = 0.9801$$

$$y'(0.2) = p_1 = p_0 + \frac{1}{6}[l_1 + 2l_2 + 2l_3 + l_4]$$

$$= 0 + \frac{1}{6}[-0.2 + 2(-0.2) + 2(-0.1980) + (-0.19616)]$$

$$= -0.19869(= -0.1987)$$

Therefore, the required solution is

$$y(0.2) = 0.9801, \quad y'(0.2) = -0.1987$$

Numerical

Analysis

Lecture 38

Numerical Analysis Lecture 39

Chapter 7

Ordinary Differential Equations

Introduction

Taylor Series

Euler Method

Runge-Kutta Method

Predictor Corrector

Method

PREDICTOR – CORRECTOR METHOD

The methods presented so far are called single-step methods, where we have seen that the computation of y at t_{n+1} that is y_{n+1} requires the knowledge of y_n only.

In predictor-corrector methods which we will discuss now, is also known as multi-step methods.

To compute the value of y at t_{n+1} , we must know the solution y at t_n , t_{n-1} , t_{n-2} , etc.

Thus, a predictor formula is used to predict the value of y at t_{n+1} and then a corrector formula is used to improve the value of y_{n+1} .

Let us consider an IVP

$$\frac{dy}{dt} = f(t, y), \quad y(t_n) = y_n$$

Using simple Euler's and modified Euler's method, we can write down a simple predictor-corrector pair (P – C) as

$$\left. \begin{aligned} P : y_{n+1}^{(0)} &= y_n + hf(t_n, y_n) \\ C : y_{n+1}^{(1)} &= y_n + \frac{h}{2} \left[f(t_n, y_n) + f(t_{n+1}, y_{n+1}^{(0)}) \right] \end{aligned} \right\}$$

Here, $y_{n+1}^{(1)}$ is the first corrected value of y_{n+1} . The corrector formula may be used iteratively as defined below:

$$y_{n+1}^{(r)} = y_n + \frac{h}{2} \left[f(t_n, y_n) + f(t_{n+1}, y_{n+1}^{(r-1)}) \right], \quad r = 1, 2, \dots$$

The iteration is terminated when two successive iterates agree to the desired accuracy

In this pair, to extrapolate the value of y_{n+1} , we have approximated the solution curve in the interval (t_n, t_{n+1}) by a straight line passing through (t_n, y_n) and (t_{n+1}, y_{n+1}) .

The accuracy of the predictor formula can be improved by considering a quadratic curve through the equally spaced points $(t_{n-1}, y_{n-1}), (t_n, y_n), (t_{n+1}, y_{n+1})$

Suppose we fit a quadratic curve of the form

$$y = a + b(t - t_{n-1}) + c(t - t_n)(t - t_{n-1})$$

where a , b , c are constants to be determined

As the curve passes through (t_{n-1}, y_{n-1}) and (t_n, y_n) and satisfies

$$\left(\frac{dy}{dt} \right)_{(t_n, y_n)} = f(t_n, y_n)$$

we obtain

$$y_{n-1} = a, \quad y_n = a + bh = y_{n-1} + bh$$

Therefore

$$b = \frac{y_n - y_{n-1}}{h}$$

and

$$\left(\frac{dy}{dt}\right)_{(t_n, y_n)} = f(t_n, y_n) = \{b + c[(t - t_{n-1}) + (t - t_n)]\}_{(t_n, y_n)}$$

Which gives

$$f(t_n, y_n) = b + c(t_n - t_{n-1}) = b + ch$$

or

$$c = \frac{f(t_n, y_n)}{h} - \frac{(y_n - y_{n-1})}{h^2}$$

Substituting these values of a , b and c into the quadratic equation, we get

$$y_{n+1} = y_{n-1} + 2(y_n - y_{n-1}) + 2[hf(t_n, y_n) - (y_n - y_{n-1})]$$

That is,

$$y_{n+1} = y_{n-1} + 2hf(t_n, y_n)$$

Thus, instead of considering the P-C pair, we may consider the P-C pair given by

$$\left. \begin{aligned} P: y_{n+1} &= y_{n-1} + 2hf(t_n, y_n) \\ C: y_{n+1} &= y_n + \frac{h}{2}[f(t_n, y_n) + f(t_{n+1}, y_{n+1})] \end{aligned} \right\}$$

The essential difference between them is, the one given above is more accurate

However, this one can not be used to predict y_{n+1} for a given IVP, because its use require the knowledge of past two points. In such a situation, a R-K method is generally used to start the predictor method.

Milne's Method

It is also a multi-step method where we assume that the solution to the given IVP is known at the past four equally spaced point t_0 , t_1 , t_2 and t_3 .

To derive Milne's predictor-corrector pair, let us consider a typical IVP

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0$$

On integration between the limits t_0 and t_4 , we get

$$\int_{t_0}^{t_4} \frac{dy}{dt} dt = \int_{t_0}^{t_4} f(t, y) dt$$

$$y_4 - y_0 = \int_{t_0}^{t_4} f(t, y) dt$$

But we know from Newton's forward difference formula

$$f(t, y) = f_0 + s\Delta f_0 + \frac{s(s-1)}{2}\Delta^2 f_0 + \frac{s(s-1)(s-2)}{6}\Delta^3 f_0 + \dots$$

where

$$s = \frac{t - t_0}{h}, \quad t = t_0 + sh$$

$$y_4 = y_0 + \int_{t_0}^{t_4} \left[f_0 + s\Delta f_0 + \frac{s(s-1)}{2}\Delta^2 f_0 + \frac{s(s-1)(s-2)}{6}\Delta^3 f_0 + \frac{s(s-1)(s-2)(s-3)}{24}\Delta^4 f_0 + \dots \right] dt$$

Now, by changing the variable of integration (from t to s), the limits of integration also changes (from 0 to 4), and thus the above expression becomes

$$y_4 = y_0 + h \int_0^4 \left[4f_0 + s\Delta f_0 + \frac{s(s-1)}{2} \Delta^2 f_0 + \frac{s(s-1)(s-2)}{6} \Delta^3 f_0 + \frac{s(s-1)(s-2)(s-3)}{24} \Delta^4 f_0 + \dots \right] ds$$

which simplifies to

$$y_4 = y_0 + h \left[4f_0 + 8\Delta f_0 + \frac{20}{3}\Delta^2 f_0 + \frac{8}{3}\Delta^3 f_0 + \frac{28}{90}\Delta^4 f_0 \right]$$

Substituting the differences

$$\Delta f_0 = f_1 - f_0, \Delta^2 f_0 = f_2 - 2f_1 + f_0,$$

It can be further simplified to

$$y_4 = y_0 + \frac{4h}{3}(2f_1 - f_2 + 2f_3) + \frac{28}{90}h\Delta^4 f_0$$

Alternatively, it can also be written as

$$y_4 = y_0 + \frac{4h}{3}(2y'_1 - y'_2 + 2y'_3) + \frac{28}{90}h\Delta^4 y'_0$$

This is known as *Milne's predictor formula*.

Similarly, integrating the original over the interval t_0 to t_2 or $s = 0$ to 2 and repeating the above steps, we get

$$y_2 = y_0 + \frac{h}{3}(y'_0 + 4y'_1 + y'_2) - \frac{1}{90}h\Delta^4 y'_0$$

which is known as *Milne's corrector formula*.

In general, Milne's predictor-corrector pair can be written as

$$\left. \begin{aligned} P : y_{n+1} &= y_{n-3} + \frac{4h}{3}(2y'_{n-2} - y'_{n-1} + 2y'_n) \\ C : y_{n+1} &= y_{n-1} + \frac{h}{3}(y'_{n-1} + 4y'_n + y'_{n+1}) \end{aligned} \right\}$$

From these equations, we observe that the magnitude of the truncation error in corrector formula is $(1/90)h\Delta^4 y'_0$, while the truncation error in predictor formula is $(28/90)\Delta^4 y'_0$.

Thus: TE in, c-formula is less than the TE in p-formula.

In order to apply this P – C method to solve numerically any initial value problem, we first predict the value of y_{n+1} by means of predictor formula, where derivatives are computed using the given differential equation itself.

Using the predicted value y_{n+1} , we calculate the derivative y'_{n+1} from the given differential equation and then we use the corrector formula of the pair to have the corrected value of y_{n+1}

This in turn may be used to obtain improved value of y_{n+1} by using corrector again.

This in turn may be used to obtain improved value of y_{n+1} by using the corrector again. This cycle is repeated until we achieve the required accuracy.

Example

Find $y(2.0)$ if $y(t)$ is the solution of

$$\frac{dy}{dt} = \frac{1}{2}(t + y)$$

$y(0) = 2$, $y(0.5) = 2.636$,
 $y(1.0) = 3.595$ and $y(1.5) = 4.968$
Use Milne's P-C method.

Solution

Taking $t_0 = 0.0$, $t_1 = 0.5$,

$t_2 = 1.0$, $t_3 = 1.5$

y_0 , y_1 , y_2 and y_3 , are given, we have to compute y_4 , the solution of the given differential equation corresponding to $t = 2.0$

The Milne's P – C pair is given as

$$P: y_{n+1} = y_{n-3} + \frac{4h}{3}(2y'_{n-2} - y'_{n-1} + 2y'_n)$$

$$C: y_{n+1} = y_{n-1} + \frac{h}{3}(y'_{n-1} + 4y'_n + y'_{n+1})$$

From the given differential equation, $y' = (t + y) / 2.$

We have,

$$y'_1 = \frac{t_1 + y_1}{2} = \frac{0.5 + 2.636}{2} = 1.5680$$

$$y'_2 = \frac{t_2 + y_2}{2} = \frac{1.0 + 3.595}{2} = 2.2975$$

$$y'_3 = \frac{t_3 + y_3}{2} = \frac{1.5 + 4.968}{2} = 3.2340$$

Now, using predictor formula, we compute

$$\begin{aligned}y_4 &= y_0 + \frac{4h}{3}(2y'_1 - y'_2 + 2y'_3) \\&= 2 + \frac{4(0.5)}{3}[2(1.5680) - 2.2975 + 2(3.2340)] \\&= 6.8710\end{aligned}$$

**Using this predicted value,
we shall compute the
improved value of y_4 from
corrector formula**

$$y_4 = y_2 + \frac{h}{3}(y'_2 + 4y'_3 + y'_4)$$

Using the available predicted value y_4 and the initial values, we compute

$$y'_4 = \frac{t_4 + y_4}{2} = \frac{2 + 6.68710}{2} = 4.4355$$

$$y'_3 = \frac{t_3 + y_3}{2} = \frac{1.5 + 4.968}{2} = 3.2340$$

and $y'_2 = 2.2975$

Thus, the first corrected value of y_4 is given by

$$\begin{aligned} y_4^{(1)} &= 3.595 + \frac{0.5}{3} [2.2975 + 4(3.234) + 4.4355] \\ &= 6.8731667 \end{aligned}$$

Suppose, we apply the corrector formula again, then we have

$$\begin{aligned} y_4^{(2)} &= y_2 + \frac{h}{3} \left[(y_2' + 4y_3' + (y_4^{(1)})') \right] \\ &= 3.595 + \frac{0.5}{3} \left[2.2975 + 4(3.234) + \frac{2 + 6.8731667}{2} \right] \\ &= 6.8733467 \end{aligned}$$

Finally, $y(2.0) = y_4 = 6.8734$.

Example

Tabulate the solution of

$$\frac{dy}{dt} = t + y, \quad y(0) = 1$$

in the interval $[0, 0.4]$ with $h = 0.1$, using Milne's P-C method.

Solution

Milne's P-C method demand the solution at first four points t_0 , t_1 , t_2 and t_3 . As it is not a self – starting method, we shall use R-K method of fourth order to get the required solution and then switch over to Milne's P – C method.

**Thus, taking $t_0 = 0$, $t_1 = 0.1$,
 $t_2 = 0.2$, $t_3 = 0.3$ we get the
corresponding y values using
R–K method of 4th order;
that is $y_0 = 1$, $y_1 = 1.1103$,
 $y_2 = 1.2428$ and $y_3 = 1.3997$
(Reference Lecture 38)**

Now, we compute

$$y'_1 = t_1 + y_1 = 0.1 + 1.1103 = 1.2103$$

$$y'_2 = t_2 + y_2 = 0.2 + 1.2428 = 1.4428$$

$$y'_3 = t_3 + y_3 = 0.3 + 1.3997 = 1.6997$$

Using Milne's predictor formula

$$\begin{aligned}P: y_4 &= y_0 + \frac{4h}{3}(2y'_1 - y'_2 + 2y'_3) \\&= 1 + \frac{4(0.5)}{3} [2(1.21103) - 1.4428 + 2(1.69971)] \\&= 1.58363\end{aligned}$$

Before using corrector formula, we compute

$$\begin{aligned} y'_4 &= t_4 + y_4(\textit{predicted value}) \\ &= 0.4 + 1.5836 = 1.9836 \end{aligned}$$

Finally, using Milne's corrector formula, we compute

$$\begin{aligned}C : y_4 &= y_2 + \frac{h}{3}(y'_4 + 4y'_3 + y'_2) \\&= 1.2428 + \frac{0.1}{3}(1.9836 + 6.7988 + 1.4428) \\&= 1.5836\end{aligned}$$

The required solution is:

t	0	0.1	0.2	0.3	0.4
y	1	1.1103	1.2428	1.3997	1.5836

Numerical Analysis Lecture 39

Numerical Analysis Lecture 40

Chapter 7

Ordinary Differential Equations

Introduction

Taylor Series

Euler Method

Runge-Kutta Method

Predictor Corrector

Method

RUNGA- KUTTA METHODS

We considered the IVP

$$\frac{dy}{dt} = f(t, y), \quad y(t_n) = y_n$$

We also defined

$$k_1 = hf(t_n, y_n), \quad k_2 = hf(t_n + \alpha h, y_n + \beta k_1)$$

**and took the weighted
average of k_1 and k_2 and
added to y_n to get y_{n+1}**

We obtained

$$W_1 + W_2 = 1, \quad W_2(\alpha f_t + \beta ff_y) = \frac{f_t + ff_y}{2}$$

Implying

$$W_1 + W_2 = 1, \quad W_2\alpha = W_2\beta = \frac{1}{2}$$

We considered two cases,

**Case I We choose $W_2 = 1/3$,
then $W_1 = 2/3$ and $\alpha = \beta = 3/2$.**

$$y_{n+1} = y_n + \frac{1}{3}(2k_1 + k_2)$$

$$k_1 = hf(t, y), \quad k_2 = hf\left(t + \frac{3}{2}h, y + \frac{3}{2}k_1\right)$$

Case II: We considered
 $W_2 = 1/2$, then $W_1 = 1/2$ and

$$\alpha = \beta = 1.$$

Then

$$y_{n+1} = y_n + \frac{k_1 + k_2}{2}$$

$$k_1 = hf(t, y), \quad k_2 = hf(t + h, y + k_1)$$

The fourth-order R-K method was described as

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

where

$$k_1 = hf(t_n, y_n)$$

$$k_2 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$

$$k_3 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right)$$

$$k_4 = hf(t_n + h, y_n + k_3)$$

PREDICTOR – CORRECTOR METHOD

Milne's Method

It is a multi-step method where we assume that the solution to the given IVP is known at the past four equally spaced point t_0 , t_1 , t_2 and t_3 .

Alternatively, it can also be written as

$$y_4 = y_0 + \frac{4h}{3}(2y'_1 - y'_2 + 2y'_3) + \frac{28}{90}h\Delta^4 y'_0$$

This is known as *Milne's predictor formula*.

Similarly, integrating the original over the interval t_0 to t_2 or $s = 0$ to 2 and repeating the above steps, we get

$$y_2 = y_0 + \frac{h}{3}(y'_0 + 4y'_1 + y'_2) - \frac{1}{90}h\Delta^4 y'_0$$

which is known as *Milne's predictor formula*.

In general, Milne's predictor-corrector pair can be written as

$$\left. \begin{aligned} P : y_{n+1} &= y_{n-3} + \frac{4h}{3}(2y'_{n-2} - y'_{n-1} + 2y'_n) \\ C : y_{n+1} &= y_{n-1} + \frac{h}{3}(y'_{n-1} + 4y'_n + y'_{n+1}) \end{aligned} \right\}$$

Adam-Moulton Method

It is another predictor-corrector method, where we use the fact that the solution to the given initial value problem is known at past four equally spaced points

$$t_n, t_{n-1}, t_{n-2}, t_{n-3}.$$

The task is to compute the value of y at t_{n+1} .

Let us consider the differential equation

$$\frac{dy}{dt} = f(t, y)$$

Integrating between the limits t_n to t_{n+1} , we have

$$\int_{t_n}^{t_{n+1}} \frac{dy}{dt} dt = \int_{t_n}^{t_{n+1}} f(t, y) dt$$

That is,

$$y_{n+1} - y_n = \int_{t_n}^{t_{n+1}} f(t, y) dt$$

To carry out integration, we proceed as follows. We employ *Newton's backward interpolation formula*, so that

$$f(t, y) = f_n + s \nabla f_n + \frac{s(s+1)}{2} \nabla^2 f_n + \frac{s(s+1)(s+2)}{6} \nabla^3 f_n + \dots$$

$$s = \frac{t - t_n}{h}$$

After substitution, we obtain

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} \left[f_n + s \nabla f_n + \frac{s(s+1)}{2} \nabla^2 f_n + \frac{s(s+1)(s+2)}{6} \nabla^3 f_n + \frac{s(s+1)(s+2)(s+3)}{24} \nabla^4 f_n + \dots \right] dt$$

Now by changing the variable of integration (from t to s), the limits of integration also changes (from 0 to 1), and thus the above expression becomes

$$y_{n+1} = y_n + h \int_0^1 \left[f_n + s \nabla f_n + \frac{s(s+1)}{2} \nabla^2 f_n + \frac{s(s+1)(s+2)}{6} \nabla^3 f_n + \frac{s(s+1)(s+2)(s+3)}{24} \nabla^4 f_n + \dots \right] ds$$

Actual integration reduces the above expression to

$$y_{n+1} = y_n + h \left(f_n + \frac{1}{2} \nabla f_n + \frac{5}{12} \nabla^2 f_n + \frac{3}{8} \nabla^3 f_n + \frac{251}{720} \nabla^4 f_n \right)$$

Now substituting the differences such as

$$\nabla f_n = f_n - f_{n-1}$$

$$\nabla^2 f_n = f_n - 2f_{n-1} + f_{n-2}$$

$$\nabla^3 f_n = f_n - 3f_{n-1} + 3f_{n-2} - f_{n-3}$$

Equation simplifies to

$$y_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) + \frac{251}{720}h\nabla^4 f_n$$

Alternatively, it can be written as

$$y_{n+1} = y_n + \frac{h}{24}[55y'_n - 59y'_{n-1} + 37y'_{n-2} - 9y'_{n-3}] + \frac{251}{720}h\nabla^4 y'_n$$

This is known as Adam's predictor formula.

The truncation error is

$$(251/720)h\nabla^4 y'_n.$$

**To obtain corrector
formula, we use Newton's
backward interpolation
formula about f_{n+1} instead
of f_n .**

We obtain

$$y_{n+1} = y_n + h \int_{-1}^0 \left[f_{n+1} + s \nabla f_{n+1} + \frac{s(s+1)}{2} \nabla^2 f_{n+1} + \frac{s(s+1)(s+2)}{6} \nabla^3 f_{n+1} + \frac{s(s+1)(s+2)(s+3)}{24} \nabla^4 f_{n+1} + \dots \right] ds$$

Carrying out the integration and repeating the steps, we get the corrector formula as

$$y_{n+1} = y_n + \frac{h}{24} (9y'_{n+1} + 19y'_n - 5y'_{n-1} + y'_{n-2}) + \left(\frac{-19}{720} \right) h \nabla^4 y'_{n+1}$$

Here, the truncation error is

$$\left(19/20\right) h \nabla^4 y'_{n+1}.$$

The truncation error in Adam's predictor is approximately thirteen times more than that in the corrector, but with opposite sign.

In general, Adam-Moulton predictor-corrector pair can be written as

$$\left. \begin{aligned} P : y_{n+1} &= y_n + \frac{h}{24} (55y'_n - 59y'_{n-1} + 37y'_{n-2} - 9y'_{n-3}) \\ C : y_{n+1} &= y_n + \frac{h}{24} (9y'_{n+1} + 19y'_n - 5y'_{n-1} + y'_{n-2}) \end{aligned} \right\}$$

Example

Using Adam-Moulton predictor-corrector method, find the solution of the initial value problem

$$\frac{dy}{dt} = y - t^2, \quad y(0) = 1$$

at $t = 1.0$, taking $h = 0.2$.

Compare it with the analytical solution.

Solution

In order to use Adam's P-C method, we require the solution of the given differential equation at the past four equally spaced points, for which we use R-K method of 4th order which is self starting.

Thus taking $t_0 = 0$, $y_0 = 1$,
 $h = 0.2$, we compute
 $k_1 = 0.2$, $k_2 = 0.218$,
 $k_3 = 0.2198$, $k_4 = 0.23596$,
and get

$$y_1 = y_0 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 1.21859$$

**Taking $t_1 = 0.2$,
 $y_1 = 1.21859$, $h = 0.2$,
we compute $k_1 = 0.23571$,
 $k_2 = 0.2492$,
 $k_3 = 0.25064$, $k_4 = 0.26184$,
and get**

$$y_2 = y_1 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 1.46813$$

**Now, we take $t_2 = 0.4$,
 $y_2 = 1.46813$, $h = 0.2$, and
compute $k_1 = 0.2616$,
 $k_2 = 0.2697$,
 $k_3 = 0.2706$, $k_4 = 0.2757$
to get**

$$y_3 = y(0.6) = y_2 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 1.73779$$

Thus, we have at our disposal

$$y_0 = y(0) = 1$$

$$y_1 = y(0, 2) = 1.21859$$

$$y_2 = y(0.4) = 1.46813$$

$$y_3 = y(0.6) = 1.73779$$

Now, we use Adam's P-C pair to calculate $y(0.8)$ and $y(1.0)$ as follows:

$$P: y_{n+1} = y_n + \frac{h}{24} (55y'_n - 59y'_{n-1} + 37y'_{n-2} - 9y'_{n-3})$$

$$C: y_{n+1} = y_n + \frac{h}{24} (9y'_{n+1} + 19y'_n - 5y'_{n-1} + y'_{n-2})$$

Thus

$$y_4^p = y_3 + \frac{h}{24} (55y'_3 - 59y'_2 + 37y'_1 - 9y'_0) \quad (1)$$

From the given differential equation, we have

$$y' = y - t^2.$$

Therefore,

$$y'_0 = y_0 - t_0^2 = 1.0$$

$$y'_1 = y_1 - t_1^2 = 1.17859$$

$$y'_2 = y_2 - t_2^2 = 1.30813$$

$$y'_3 = y_3 - t_3^2 = 1.37779$$

Hence, from Eq. (1), we get

$$y(0.8) = y_4^p = 1.73779 + \frac{0.2}{24} (75.77845 - 77.17967 + 43.60783 - 9) \\ = 2.01451$$

Now to obtain the corrector value of y at $t = 0.8$, we use

$$y_4^c = y^c(0.8) = y_3 + \frac{h}{24} (9y_4' + 19y_3' - 5y_2' + y_1') \quad (2)$$

But,

$$9y'_4 = 9(y_4^p - t_4^2) = 9[2.01451 - (0.8)^2] = 12.37059$$

Therefore,

$$y_4 = y^c(0.8) = 1.73779 + \frac{0.2}{24}(12.37059 + 26.17801 - 6.54065 + 1.17859)$$

$$= 2.01434 \quad (3)$$

Proceeding similarly, we get

$$y_5^p = y^p(1.0) = y_4 + \frac{h}{24}(55y'_4 - 59y'_3 + 37y'_2 - 9y'_1)$$

**Noting that $y'_4 = y_4 - t_4^2 = 1.3743$,
we calculate**

$$y_5^p = 2.01434 + \frac{0.2}{24} (75.5887 - 81.28961 + 48.40081 - 10.60731) \\ = 2.28178$$

**Now, the corrector formula
for computing y_5 is given by**

$$y_5^c = y^c(1.0) = y_4 + \frac{h}{24} (9y'_5 + 19y'_4 - 5y'_3 + y'_2) \quad (4)$$

But, $9y'_5 = 9\left(y_5^p - t_5^2\right) = 11.53602$

Thus, finally we get

$$y_5 = y(1.0) = 2.01434 + \frac{0.2}{24}(11.53602 + 26.17801 - 6.54065 + 1.17859)$$

$$= 2.28393$$

(5)

The analytical solution can be seen in the following steps.

$$\frac{dy}{dt} - y = -t^2$$

After finding integrating factor and solving, we get

$$\frac{d}{dt} ye^{-t} = -e^{-t} t^2$$

Integrating, we get

$$ye^{-t} = -\int e^{-t} t^2 dt = \int t^2 d(e^{-t}) = t^2 e^{-t} + 2te^{-t} + c$$

That is, $y = t^2 + 2t + 2 + \frac{c}{e^{-t}}$

Now using the initial condition, $y(0) = 1$, we get $c = -1$.

Therefore, the analytical solution is given by

$$y = t^2 + 2t + 2 - e^t$$

from which, we get

$$y(1.0) = 5 - e = 2.2817$$

Convergence and Stability Considerations

**The numerical solution
of a differential equation
can be shown to
converge to its exact
solution, if the step size
 h is very small.**

The numerical solution of a differential equation is said to be stable if the error do not grow exponentially as we compute from one step to another.

Stability consideration are very important in finding the numerical solutions of the differential equations either by *single-step* methods or by using *multi-step* methods.

However, theoretical analysis of stability and convergence of R -K methods and P –C methods are highly involved and obtain numerically stable solution using 4th order R – K method to the simple problem $y' = Ay$ gives us stability condition as $-2.78 \leq Ah$

In practice, to get numerically stable solutions to similar problems, we choose the value of h much smaller than the value given by the above condition and also check for consistency of the result.

Another topic of interest which is not considered, namely the stiff system of differential equations that arises in many chemical engineering systems, such as chemical reactors, where the rate constants for the reactions involved are widely different.

Most of the realistic stiff DE do not have analytical solutions and therefore only numerical solutions can be obtained.

However, to get numerically stable solutions, a very small step size h is required, to use either R-K methods or P – C methods.

More computer time is required

Numerical Analysis Lecture 40

Numerical Analysis Lecture 41

Examples of Differential Equations

Recall EULER METHOD

We considered the differential equation of first order with the initial condition $y(t_0) = y_0$.

$$\frac{dy}{dt} = f(t, y)$$

**We obtained the solution
of the given differential
equation in the form of a
recurrence relation**

$$y_{m+1} = y_m + hf(t_m, y_m)$$

In fact Euler's method constructs $w_i \sim y(t_i)$ for each $i = 0, 1, \dots, N-1$ by deleting the remainder term. Thus the Euler's Method is

$$w_0 = \alpha,$$

$$w_{i+1} = w_i + hf(t_i, w_i)$$

for each $i = 0, 1, \dots, N-1$

Euler's algorithm

Let us try to approximate the solution of the given IVP at $(N+1)$ equally spaced numbers in the interval $[a, b]$

$$y' = f(t, y),$$

$$a \leq t \leq b, \quad y(a) = \alpha$$

INPUT *endpoints a, b ; integer N , initial condition (alpha)*

OUTPUT *approximate 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 $i = 0, 1, \dots, N$ *do* Step 3, 4.

Step 3

Set $w = w + h f(t, w)$; (compute w_i).

$t = a + i h$ (compute t_i)

Step 4 OUTPUT (t, w)

Step 5 STOP

Example

Use Euler's method to approximate the solution of IVP

$$y' = y - t^2 + 1, \quad 0 < t < 2,$$

$$y(0) = 0.5 \quad \text{with } N = 10.$$

Solution

Here, $h = 0.2$, $t_i = 0.2i$, $w_0 = 0.5$

and $w_{i+1} = w_i + h (w_i - t_i^2 + 1)$

$$= w_i + 0.2[w_i - 0.04i^2 + 1]$$

$$= 1.2 w_i - 0.008i^2 + 0.2$$

for $i = 0, 1, \dots, 9$.

The exact solution is

$$y(t) = (t+1)^2 - 0.5 e^t$$

> alg051();

This is Euler's Method.

**Input the function $F(t,y)$ in
terms of t and y**

For example: $y-t^2+1$

> $y-t^2+1$

**Input left and right endpoints separated
by blank**

> 0 2

Input the initial condition

> 0.5

**Input a positive integer for the number
of subintervals**

> 10

Choice of output method:

- 1. Output to screen**
- 2. Output to text file**

Please enter 1 or 2

> 1

Output

t

w

0.000 0.50000000

0.200 0.80000000

0.400 1.15200000

0.600 1.55040000

0.800 1.98848000

1.000 2.45817600

> alg051();

This is Euler's Method.

**Input the function $F(t,y)$ in
terms of t and y**

For example: $y-3*t^2+4$

> $y-3*t^2+4$

**Input left and right hand points separated
by a blank**

>0 1

Input the initial condition

> 0.5

**Input a positive integer for the number of
subintervals**

> 10

Choice of output method:

1. Output to screen

2. Output to text file

Please enter 1 or 2

> 1

Output

t	w
0.000	0.50000000
0.100	0.95000000
0.200	1.44200000
0.300	1.97420000
0.400	2.54462000
0.500	3.15108200
0.600	3.79119002
0.700	4.46230992
0.800	5.16154001
0.900	5.88569442
1.000	6.63126336

Recall

Runge-Kutta

(Order Four)

METHOD

The fourth-order R-K method was described as

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

where

$$k_1 = hf(t_n, y_n)$$

$$k_2 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$

$$k_3 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right)$$

$$k_4 = hf(t_n + h, y_n + k_3)$$

Example

Solve the following differential equation

$$\frac{dy}{dt} = t + y$$

with the initial condition $y(0) = 1$,
using fourth- order Runge-Kutta
method from $t = 0$ to $t = 0.4$
taking $h = 0.1$

Solution

The fourth-order Runge-Kutta method is described as

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (1)$$

where

$$k_1 = hf(t_n, y_n)$$

$$k_2 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$

$$k_3 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right)$$

$$k_4 = hf(t_n + h, y_n + k_3)$$

In this problem,

$$f(t, y) = t + y, \quad h = 0.1, \quad t_0 = 0, \quad y_0 = 1.$$

As a first step, we calculate

$$k_1 = hf(t_0, y_0) = 0.1(1) = 0.1$$

$$\begin{aligned} k_2 &= hf(t_0 + 0.05, y_0 + 0.05) \\ &= hf(0.05, 1.05) = 0.1[0.05 + 1.05] \\ &= 0.11 \end{aligned}$$

$$\begin{aligned} k_3 &= hf(t_0 + 0.05, y_0 + 0.055) \\ &= 0.1(0.05 + 1.055) \\ &= 0.1105 \end{aligned}$$

$$k_4 = 0.1(0.1 + 1.1105) = 0.12105$$

Now, we compute from

$$\begin{aligned} y_1 &= y_0 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ &= 1 + \frac{1}{6}(0.1 + 0.22 + 0.2210 + 0.12105) \\ &= 1.11034 \end{aligned}$$

Therefore $y(0.1) = y_1 = 1.1103$

In the second step, we have to find $y_2 = y(0.2)$

We compute

$$k_1 = hf(t_1, y_1) = 0.1(0.1 + 1.11034) = 0.121034$$

$$k_2 = hf\left(t_1 + \frac{h}{2}, y_1 + \frac{k_1}{2}\right)$$

$$= 0.1[0.15 + (1.11034 + 0.060517)] = 0.13208$$

$$k_3 = hf\left(t_1 + \frac{h}{2}, y_1 + \frac{k_2}{2}\right)$$

$$= 0.1[0.15 + (1.11034 + 0.06604)] = 0.132638$$

$$k_4 = hf(t_1 + h, y_1 + k_3)$$

$$= 0.1[0.2 + (1.11034 + 0.132638)] = 0.1442978$$

From Equation (1), we see that

$$y_2 = 1.11034 + \frac{1}{6}[0.121034 + 2(0.13208) + 2(0.132638) + 0.1442978] = 1.2428$$

Similarly we calculate,

$$k_1 = hf(t_2, y_2) = 0.1[0.2 + 1.2428] = 0.14428$$

$$k_2 = hf\left(t_2 + \frac{h}{2}, y_2 + \frac{k_1}{2}\right) = 0.1[0.25 + (1.2428 + 0.07214)] = 0.156494$$

$$k_3 = hf\left(t_1 + \frac{h}{2}, y_1 + \frac{k_2}{2}\right) = 0.1[0.3 + (1.2428 + 0.078247)] = 0.1571047$$

$$k_4 = hf(t_2 + h, y_2 + k_3) = 0.1[0.3 + (1.2428 + 0.1571047)] = 0.16999047$$

Using equation (1), we compute

$$y(0.3) = y_3 = y_2 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) = 1.399711$$

Finally, we calculate

$$k_1 = hf(t_3, y_3) = 0.1[0.3 + 1.3997] = 0.16997$$

$$k_2 = hf\left(t_3 + \frac{h}{2}, y_3 + \frac{k_1}{2}\right) = 0.1[0.35 + (1.3997 + 0.084985)] = 0.1834685$$

$$k_3 = hf\left(t_3 + \frac{h}{2}, y_3 + \frac{k_2}{2}\right) = 0.1[0.35 + (1.3997 + 0.091734)] = 0.1841434$$

$$k_4 = hf(t_3 + h, y_3 + k_3) = 0.1[0.4 + (1.3997 + 0.1841434)] = 0.19838434$$

**Using them in equation (1),
we get**

$$\begin{aligned}y(0.4) &= y_4 \\&= y_3 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\&= 1.58363\end{aligned}$$

which is the required result.

Runge-Kutta Order Four

$$w_0 = \alpha$$

$$k_1 = hf(t_i, w_i)$$

$$k_2 = hf\left(t_i + \frac{h}{2}, w_i + \frac{k_1}{2}\right)$$

$$k_3 = hf\left(t_i + \frac{h}{2}, w_i + \frac{k_2}{2}\right)$$

$$k_4 = hf(t_i + h, w_i + k_3)$$

$$w_{i+1} = w_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

(1)

RK4 algorithm

Let us try to approximate the solution of the given IVP at $(N+1)$ equally spaced numbers in the interval $[a, b]$

$$y' = f(t, y),$$

$$a \leq t \leq b, \quad y(a) = \alpha$$

INPUT *endpoints a, b ; integer N , initial condition (alpha)*

OUTPUT *approximate 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 $i = 0, 1, \dots, N$ *do* Step 3 - 5.

Step 3

Set

$$K_1 = hf(t, w)$$

$$K_2 = hf\left(t + \frac{h}{2}, w + \frac{K_1}{2}\right)$$

$$K_3 = hf\left(t + \frac{h}{2}, w + \frac{K_2}{2}\right)$$

$$K_4 = hf(t + h, w + K_3)$$

$$w_{i+1} = w_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Step 3

Set $w = w + h f(t, w)$; (compute w_i).

$t = a + i h$ (compute t_i)

Step 4 OUTPUT (t, w)

Step 5 STOP

> alg052();

**This is the Runge-Kutta
Order Four Method.**

**Input the function $F(t,y)$ in
terms of t and y**

For example: $y-t^2+1$

> $y-t^2+1$

**Input left and right endpoints
separated by blank**

> 0 2

Input the initial condition

> 0.5

**Input a positive integer for the
number of subintervals**

> 10

Choice of output method:

- 1. Output to screen**
- 2. Output to text file**

Please enter 1 or 2

> 1

Output

t	w
0.000	0.50000000
0.200	0.8292933
0.400	1.2140762
0.600	1.6489220
0.800	2.1272027
1.000	2.6408227
1.200	3.1798942
1.400	3.7323401
1.600	4.2834095
1.800	4.8150857
2.000	5.3053630

Numerical Analysis Lecture 41

Numerical Analysis Lecture 42

Examples of Numerical Differentiation

The simplest formula for differentiation is

$$f'(x_0) = \frac{1}{h} [f(x_0 + h) - f(x_0)]$$

$$-\frac{h^2}{2} f^{(2)}(\xi),$$

Example

Let $f(x) = \ln x$ and $x_0 = 1.8$.

Then the quotient

$$\frac{f(1.8 + h) - f(1.8)}{h}, \quad h > 0,$$

is used to approximate $f'(1.8)$

with error $\frac{|hf''(\xi)|}{2} = \frac{|h|}{2\xi^2} \leq \frac{|h|}{2(1.8)^2},$

where

$$1.8 < \xi < 1.8 + h.$$

**Let us see the results for
 $h = 0.1, 0.01, \text{ and } 0.001.$**

h	$f(1.8 + h)$	$\frac{f(1.8 + h) - f(1.8)}{h}$	$\frac{ h }{2(1.8)^2}$
0.1	0.64185389	0.5406722	0.0154321
0.01	0.59332685	0.5540180	0.0015432
0.001	0.58834207	0.5554013	0.0001543

Since $f'(x) = 1/x$,

**The exact value of $f'(1.8)$
is $0.55\bar{5}$**

**and the error bounds
are appropriate.**

The following two **three point** formulas become especially useful if the nodes are equally spaced, that is, when

$$x_1 = x_0 + h \text{ and}$$

$$x_2 = x_0 + 2h,$$

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

**where ξ_0 lies between x_0 and $x_0 + 2h$,
and**

$$f'(x_0) = \frac{1}{2h} [f(x_0 + h) - f(x_0 - h)]$$

$$- \frac{h^2}{6} f^{(3)}(\xi_1),$$

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

Given in Table below are values for $f(x) = xe^x$.

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

Since

$$f'(x) = (x+1)e^x, f'(2.0) = 22.167168.$$

**Approximating $f'(2.0)$
using the various
three-and five-point formulas
produces the following results.**

Three point formulas:

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

$$+ \frac{h^2}{3} f^{(3)}(\xi_0),$$

$$f'(x_0) = \frac{1}{2h} [f(x_0 + h) - f(x_0 - h)]$$

$$- \frac{h^2}{6} f^{(3)}(\xi_1),$$

Using three point formulas

we get

$$h = 0.1 : \frac{1}{0.2} [-3f(2.0) + 4f(2.1) - f(2.2)]$$

$$= 22.032310,$$

$$h = -0.1 : \frac{1}{-0.2} [-3f(2.0) + 4f(1.9) - f(1.8)]$$

$$= 22.0054525,$$

$$h = 0.1 : \frac{1}{0.2} [f(2.1) - f(1.9)] \\ = 22.228790,$$

$$h = 0.2 : \frac{1}{0.4} [f(2.2) - f(1.8)] \\ = 22.414163.$$

Five point formula

Using the five point formula with $h = 0.1$ (the only formula applicable):

$$\begin{aligned} f'(x_0) &= \frac{1}{12h} \left[f(x_0 - 2h) - 8f(x_0 - h) + \right. \\ &\quad \left. 8f(x_0 + h) - f(x_0 + 2h) \right] \\ &= \frac{1}{0.2} [f(1.8) - 8f(1.9) + 8f(2.1) - f(2.2)] \\ &= 22.166999. \end{aligned}$$

The errors in the formulas are approximately

$$1.35 \times 10^{-1}, 1.13 \times 10^{-1}, -6.16 \times 10^{-2}, \\ -2.47 \times 10^{-1},$$

and $1.69 \times 10^{-4},$

respectively. Clearly, the five-point formula gives the superior result.

Consider approximating $f'(0.900)$ for $f(x) = \sin x$, using the values in table [the true value is $\cos(0.900) = 0.62161$.]

x	$\sin x$	x	$\sin x$
0.800	0.71736	0.901	0.78395
0.850	0.75128	0.902	0.78457
0.880	0.77074	0.905	0.78643
0.890	0.77707	0.910	0.78950
0.895	0.78021	0.920	0.79560
0.898	0.78208	0.950	0.81342
0.899	0.78270	1.000	0.84147

Using the formula

$$f'(0.900) \approx \frac{f(0.900 + h) - f(0.900 - h)}{2h}$$

with different values of h gives the approximations in table given below:

h	Approximation to $f'(0.900)$	Error
0.001	0.62500	0.00339
0.002	0.62250	0.00089
0.005	0.62200	0.00039
0.010	0.62150	-0.00011
0.020	0.62150	-0.00011
0.050	0.62140	-0.00021
0.100	0.62055	-0.00106

Examples of Numerical Integration

EXAMPLE

The Trapezoidal rule for a function f on the interval $[0, 2]$ is

$$\int_0^2 f(x)dx = \frac{h}{2}[f(x_0) + f(x_1)]$$

$$\int_0^2 f(x)dx \approx f(0) + f(2),$$

while Simpson's rule for f on $[0, 2]$ is

$$\int_0^2 f(x)dx = \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)].$$

That is

$$\int_0^2 f(x)dx \approx \frac{1}{3}[f(0) + 4f(1) + f(2)].$$

$f(x)$	x^2	x^4	$1/(x+1)$	$\sqrt{1+x^2}$	$\sin x$	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

**Use close and open
formulas listed below to
approximate**

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

Some of the common closed Newton-Cotes formulas with their error terms are as follows:

n = 1: Trapezoidal rule

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

where $x_0 < \xi < x_1.$

n = 2: Simpson's rule

$$\int_{x_0}^{x_2} f(x)dx = \frac{h}{2}[f(x_0) + 4f(x_1) + f(x_2)] - \frac{h^5}{90} f^{(4)}(\xi),$$

$$x_0 < \xi < x_2.$$

n = 3: Simpson's rule

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

$$x_0 < \xi < x_3.$$

n = 4:

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

where $x_0 < \xi < x_4$.

n = 0: Midpoint rule

$$\int_{x_2}^{x_1} f(x)dx = 2hf(x_0) + \frac{h^3}{3}f''(\xi), \quad \textbf{where} \quad x_{-1} < \xi < x_1.$$

n		1	2	3	4
Closed formulas		0.27768018	0.29293264	0.29291070	0.29289318
Error		0.01521303	0.00003942	0.00001748	0.00000004
Open formulas		0.29798754	0.29285866	0.29286923	
Error		0.00509432	0.00003456	0.00002399	

Composite Numerical Integration

EXAMPLE 1

Consider approximating $\int_0^{\pi} \sin x dx$
with an absolute error less than
0.00002, using the Composite
Simpson's rule. The Composite
Simpson's rule gives

$$\int_0^{\pi} \sin x dx = \frac{h}{3} \left[2 \sum_{j=1}^{(n/2)-1} \sin x_{2j} + 4 \sum_{j=1}^{n/2} \sin x_{2j-1} \right] - \frac{\pi h^4}{180} \sin \mu.$$

Since the absolute error is required to be less than 0.00002, the inequality

$$\left| \frac{\pi h^4}{180} \sin \mu \right| \leq \frac{\pi h^4}{180} = \frac{\pi^5}{180 n^4} < 0.00002$$

is used to determine n and h . Computing these calculations gives n greater than or equal to 18. If $n = 20$, then the formula becomes

$$\int_0^{\pi} \sin x dx \approx \frac{\pi}{60} \left[2 \sum_{j=1}^9 \sin \left(\frac{j\pi}{10} \right) + 4 \sum_{j=1}^{10} \sin \left(\frac{(2j-1)\pi}{20} \right) \right] = 2.000006.$$

To be assured of this degree of accuracy using the Composite Trapezoidal rule requires that

$$\left| \frac{\pi h^2}{12} \sin \mu \right| \leq \frac{\pi h^2}{12} = \frac{\pi^3}{12n^2} < 0.00002$$

or that $n \geq 360$. Since this is many more calculations than are needed for the Composite Simpson's rule, it is clear that it

would be undesirable to use the Composite Trapezoidal rule on this problem. For comparison purposes, the Composite Trapezoidal rule with $n = 20$ and $h = \pi/20$ gives

$$\int_0^{\pi} \sin x dx \approx \frac{\pi}{40} \left[2 \sum_{j=1}^{19} \sin \left(\frac{j\pi}{20} \right) + \sin 0 + \sin \pi \right]$$
$$= \frac{\pi}{40} \left[2 \sum_{j=1}^{19} \sin \left(\frac{j\pi}{20} \right) \right] = 1.9958860.$$

The exact answer is 2; so Simpson's rule with $n = 20$ gave an answer well within the required error bound, whereas the Trapezoidal rule with $n = 20$ clearly did not.

An Example of Industrial applications: A company advertises that every roll of toilet paper has at least 250 sheets. The probability that there are 250 or more sheets in the toilet paper is given by

$$P(y \geq 250) = \int_{250}^{\infty} 0.3515 e^{-0.3881(y-252.2)^2} dy$$

Approximating the above integral as

$$P(y \geq 250) = \int_{250}^{270} 0.3515 e^{-0.3881(y-252.2)^2} dy$$

- a) use single segment Trapezoidal rule to find the probability that there are 250 or more sheets.**
- b) Find the true error, E_t for part (a).**
- C) Find the absolute relative true error for part (a).**

$$I \approx (b-a) \left[\frac{f(a) + f(b)}{2} \right] , \text{ where}$$

$$a = 250 \quad b = 270$$

$$f(y) = 0.3515e^{-0.3881(y-252.2)^2} \quad f(250) = 0.3515e^{-0.3881(250-252.2)^2} \\ = 0.053721$$

$$f(270) = 0.3515e^{-0.3881(270-252.2)^2} = 1.3888 \times 10^{-54}$$

$$I = (270 - 250) \left[\frac{0.053721 + 1.3888 \times 10^{-54}}{2} \right]$$

$$= 0.53721$$

b) The exact value of the above integral cannot be found. We assume the value obtained by adaptive numerical integration using Maple as the exact value for calculating the true error and relative true error.

$$P(y \geq 250) = \int_{250}^{270} 0.3515 e^{-0.3881(y-252.2)^2} dy$$
$$= 0.97377$$

so the true error is

$$= 0.97377 - 0.53721$$
$$= 0.43656$$

The absolute relative true error,

$|\epsilon_t|$, would then be

$$|\epsilon_t| = \left| \frac{\text{True Error}}{\text{True Value}} \right| \times 100$$

$$= \left| \frac{0.97377 - 0.53721}{0.97377} \right| \times 100$$

$$= 44.832\%$$

Improper Integrals

EXAMPLE

To approximate the values of the improper integral

$$\int_0^1 \frac{e^x}{\sqrt{x}} dx,$$

we will use the Composite Simpson's rule with $h = 0.25$. Since the fourth Taylor polynomial for e^x about $x = 0$ is

$$P_4(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24}.$$

We have

$$\int_0^1 \frac{P_4(x)}{\sqrt{x}} dx = \lim_{M \rightarrow 0^+} \left[2x^{1/2} + \frac{2}{3}x^{3/2} + \frac{1}{5}x^{5/2} + \frac{1}{21}x^{7/2} + \frac{1}{180}x^{9/2} \right]_M^1$$

$$= 2 + \frac{2}{3} + \frac{1}{5} + \frac{1}{21} + \frac{1}{108} \approx 2.9235450.$$

Table below lists the approximate values of

$$G(x) = \begin{cases} \frac{e^x - P_4(x)}{\sqrt{4}} & \text{when } 0 < x \leq 1 \\ 0, & \text{when } x = 0 \end{cases}$$

x	$G(x)$
0.00	0
0.25	0.0000170
0.50	0.0004013
0.75	0.0026026
1.00	0.0099485

Applying the Composite Simpson's rule to G using these data gives

$$\begin{aligned}\int_0^1 G(x)dx &\approx \\ &\frac{0.25}{3}[0 + 4(0.0000170) + 2(0.0004013) \\ &+ 4(0.0026026) + 0.0099485] \\ &= 0.0017691\end{aligned}$$

Hence

$$\int_0^1 \frac{e^x}{\sqrt{x}} dx \approx 2.9235450 + 0.0017691 = 2.9253141.$$

This result is accurate within the accuracy of the Composite Simpson's rule approximation for the function G . Since $|G^{(4)}(x)| < 1$

on $[0, 1]$, the error is bounded by
$$\frac{1-0}{180} (0.25)^4 (1) = 0.0000217.$$

EXAMPLE

To approximate the value of the improper integral

$$I = \int_1^{\infty} x^{-3/2} \sin \frac{1}{x} dx.$$

we make the change of variable $t = x^{-1}$ to obtain

$$I = \int_0^1 t^{1/2} \sin t \, dt.$$

The fourth Taylor polynomial, $P_4(t)$, for $\sin t$ about 0 is

$$P_4(t) = t - \frac{1}{6}t^3,$$

So we have

$$\begin{aligned} I &= \int_0^1 \frac{\sin t - t + \frac{1}{6}t^3}{t^{1/2}} dt + \int_0^1 t^{1/2} - \frac{1}{6}t^{5/2} dt \\ &= \int_0^1 \frac{\sin t - t + \frac{1}{6}t^3}{t^{1/2}} dt + \left[\frac{2}{3}t^{3/2} - \frac{1}{21}t^{7/2} \right]_0^1 \\ &= \int_0^1 \frac{\sin t - t + \frac{1}{6}t^3}{t^{1/2}} dt + 0.61904761. \end{aligned}$$

Applying the Composite Simpson's rule with $n = 8$ to the remaining integral gives

$$I = 0.0014890097 + 0.61904761 = 0.62053661,$$

which is accurate to within 4.0×10^{-8} .

Numerical Analysis

Lecture 42

Numerical Analysis Lecture 43

An Introduction to MAPLE

Maple is a comprehensive computer system for advanced mathematics.

It includes facilities for interactive algebra, calculus, discrete mathematics, graphics, numerical computation etc.

It provides a unique environment for rapid development of mathematical programs using its vast library of built-in functions and operations.

Syntax :As with any computer language, Maple has its own syntax.

We try to explain some of the symbols used in Maple

Symbol	Description	Examples	Sample Output
;	End-of-line. Tells Maple to process the line and show the output.	hello;	hello
:	End-of-line. Tells Maple to process the line and hide the output.	hello:	
:=	Assignment. Lets you assign values to variables.	a := 3; a;	a := 3 3
+, -	Addition, subtraction.	1 + 3; 1 - 3;	4 -2
*, /	Multiplication, division	3*412; 1236/3; 7/3;	1236 412 7/3

^, sqrt	Power, square root	2^3; sqrt(2); 2^(1/2);	8 $\sqrt{2}$ $\sqrt{2}$
evalf, .	Floating-point (decimal) evaluation	evalf(7/3) ; 7.0/3;	2.333333333 3 2.333333333 3
I,Pi	Imaginary unit, Pi.	2 + 3*I; (2*I)^2; evalf(Pi);	2+3I -4 3.14159265
%, %%	Recall the last output, recall the second-to-last output, etc.	%; %%%;	3.14159265 -4

Some syntactical Tips:

Maple is case sensitive. foo, Foo, and FOO are three different things.

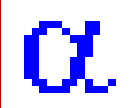
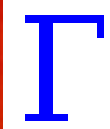
**$x*y$ gives the product of x and y ,
 xy is one variable**

To get the constant e use $\exp(1)$.

Using the % operator can give confusing results. It always returns the last output from the Kernel, which may have nothing to do with where the cursor is (or which worksheet is active).

If Maple doesn't recognize something, it assumes it as a variable; e.g. typing i^2 will give you i^2 , while we may be wanted -1 .

Spaces are optional.

Greek letters may be entered by spelling their name. For example, alpha is always displayed as , and Gamma is displayed as  (note upper-case).

Built-in Data Capabilities

Maple can handle arbitrary-precision floating point numbers. In other words, Maple can store as many digits for a number as you like, up to the physical limits of your computer's memory. To control this, use the Digits variable.

```
sqrt(2.0);
```

```
1.414213562
```

```
Digits := 20;
```

```
sqrt(2.0);
```

```
1.4142135623730950488
```

Maple sets Digits to be 10 by default. You can also temporarily get precision results by calling evalf with a second argument.

```
evalf(sqrt(2), 15);
```

1.41421356237310

Large integers are handled automatically

Using symbolic computation

**The main feature of Maple is
symbolic computation.**

**In other words, Maple does
algebra.**

Example	Output	Comments
$(x + y)^2;$	$(x + y)^2$	A basic expression.
$k := x*y + y^2;$	$k := xy + y^2$	k is now an alias for the expression. Note that k is simply another name for the expression - they are not equal in the mathematical sense.
$p := k/(x+y);$	$\frac{1}{3}x^3 - 4x^2 + 15x$	You can now use k to refer to the expression. Maple immediately substitutes the value of k.

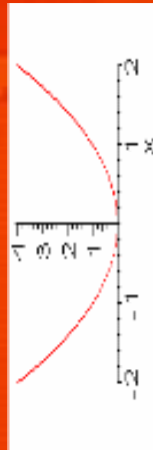
<code>k := 'k';</code>	<code>k</code>	You can unassign a variable by assigning it to its own name in single quotes.
<code>simplify(p);</code>	<code>y</code>	The simplify command does algebraic simplification.
<code>p := x^2 - 8*x + 15;</code>	<code>p := x^2 - 8x + 15</code>	Maple doesn't mind if you re-use names. The old value is lost.
<code>solve(p=3,x);</code>	<code>2,6</code>	Use the solve command to solve equations. Note the use of the = sign. Here, it is used in a mathematical sense. Maple will try different values for x until it finds all of them that make the mathematical statement $x^2 - 8x + 15 = 3$ true.

```
dpdx :=  
diff(p,x);
```

```
dpdx :  
= 2x -  
8
```

**The diff command
differentiates an
expression with
respect to a variable.**

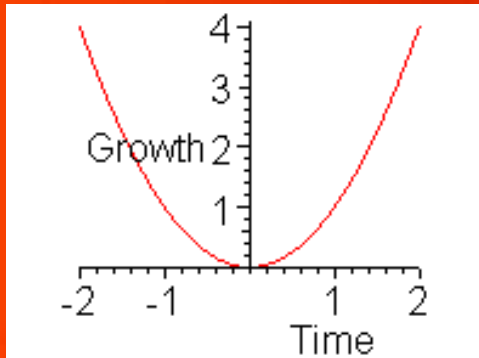
```
int(p,x);
```



**The int command
integrates an
expression.
Note that the
constant of
integration is left off.**

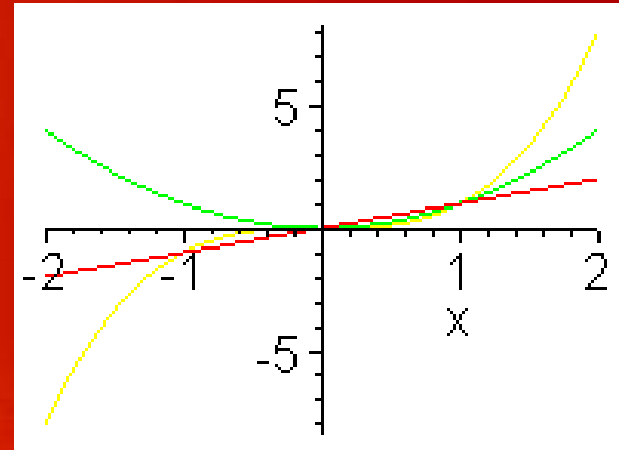
Basic Plotting

Maple can produce graphs very easily. Here are some examples, showcasing the basic capabilities.



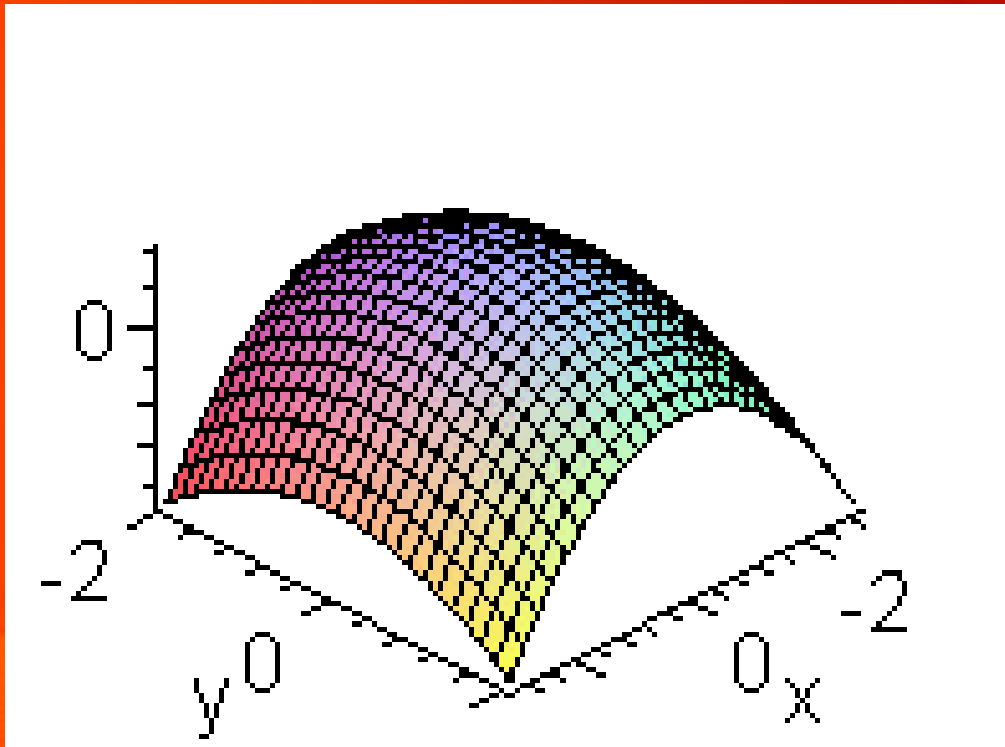
```
plot( x^2, x=-2..2);
```

A basic plot.

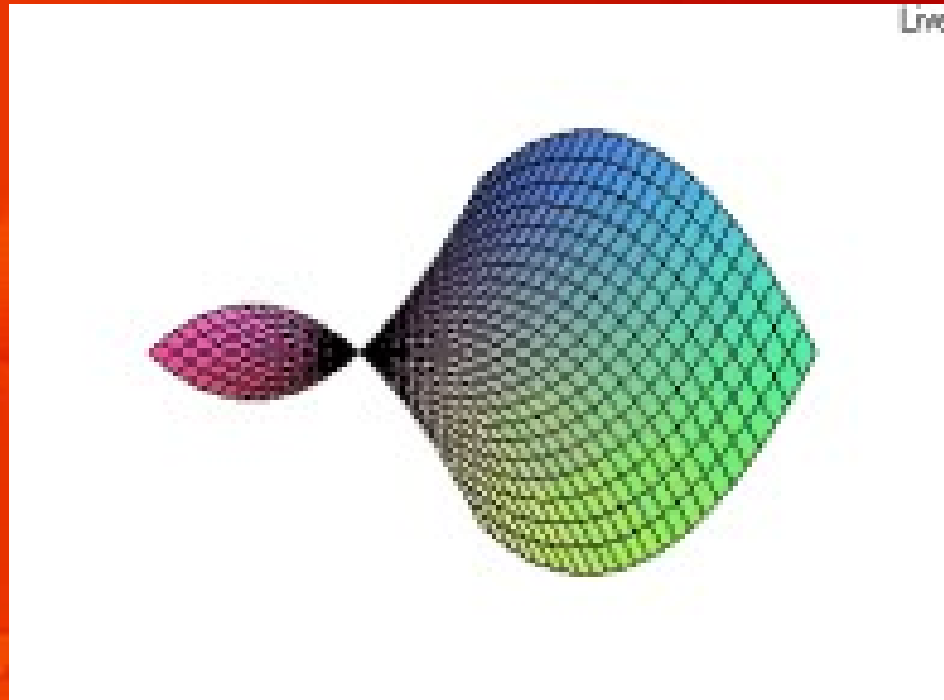


```
plot( x^2, x=-2..2, y=-10..10);
```

**A plot with
vertical axis
control.**



`plot([x, x^2, x^3], x=-2..2);`
**Plot multiple expressions
by enclosing them in
brackets.**



```
plot3d(4-x^2-y^2, x=-3..3, y=-  
2..2);
```

A basic 3-d plot.

`smartplot3d(x^2-y^2);`
**Using smartplot to let
maple set it's own
scaling.**

**Eigenvals and vectors of a
numeric matrix :**

Calling Sequence

Eigenvals(A, vecs)

Eigenvals(A, B, vecs)

Parameters

A,B - square matrices
of real or complex numbers

vecs - (optional) name
to be assigned the matrix
of eigenvectors

Example

> A := array([[1,2,4],[3,7,2],[5,6,9]]);

$$A := \begin{bmatrix} 1 & 2 & 4 \\ 3 & 7 & 2 \\ 5 & 6 & 9 \end{bmatrix}$$

> evalf(Eigenvals(A));

[-.8946025434, 13.74788901, 4.146713483]

> lambda :=

evalf(Eigenvals(A,vecs));

$\lambda :=$ [-.8946025434, 13.74788901, 4.146713483]

> print(vecs);

0.946575123	0.3592673648	-.2987282688
-.2818195168	0.4346670837	0.7371720079
-.3074361518	0.9276415841	-.6035890728

linalg[eigenvectors] - find the eigenvectors of a matrix

Calling Sequence

eigenvectors(A)

eigenvectors(A, 'radical')

eigenvectors(A, 'implicit')

Parameters

A - square matrix

**The command
with(linalg,eigenvectors)
allows the use of the
abbreviated form of this
command.**

> with(linalg):

**Warning, the protected
names norm and trace have
been redefined and
unprotected**

**> A := matrix(3,3, [1,-3,3,3, -
5,3,6,-6,4]);**

$$A := \begin{bmatrix} 1 & -3 & 3 \\ 3 & -5 & 3 \\ 6 & -6 & 4 \end{bmatrix}$$

```
> e := eigenvalues(A);
```

```
e := 4, -2, -2
```

```
> v := [eigenvectors(A)];
```

```
v := [[4, 1, {[1, 1, 2]}], [-2, 2, {[1, 1, 0], [-1, 0, 1]}]]
```

> v[1][1]; # The first eigenvalue

1

> v[1][2]; # Its multiplicity

1

> v[1][3]; # Its eigenvectors

{[1, 1, 2]}

> v[2][1]; # The second eigenvalue

-2

> v[2][2]; # Its multiplicity

> v[2][2]; # Its multiplicity



Help

its worksheet interface

- **Waiting for command;**
- **Restart; refresh memory;**
- **# #; comments so no
action implied**

Eval - Evaluate an expression

Calling Sequence

Eval(a, x=n))

Eval(a, {x1=n1,x2=n2,...})

Parameters

a - an expression

x, x1, x2,... - names

n, n1, n2,... - evaluation points

Description

The Eval function is a place holder for evaluation at a point.

The expression a is evaluated at $x = n$ ($x_1=n_1, x_2=n_2, \dots$ for the multivariate case).

The call $\text{Eval}(a, x=n) \bmod p$ evaluates the polynomial a at $x=n$ modulo p .

Note: The polynomial must be a multivariate polynomial over a finite field.

The call `modp1(Eval(a,n),p)` evaluates the polynomial `a` at $x = n$ modulo p where `a` must be a univariate polynomial in the `modp1` representation, with `n` an integer and `p` an integer > 1 .

Examples

➤ $\text{Eval}(x^2+1, x=3) \bmod 5;$

0

➤ $\text{Eval}(x^2+y, \{x=3, y=2\}) \bmod 5;$

1

> Eval (int (f(x),x), x=y);

$$\int f(x) dx \Big|_{x=y}$$

Eigen values ?;

Solution of Problems

We can use Maple For:

Solution of non-linear equations

by Newton's Method

by Bisection Method

**Solution of System of linear
equations.**

Numerical Integration.

Numerical Solution of ODE's.

Maple performs both numerical and symbolic integration.

Please note that the Maple uses the int function for the both numerical and symbolic integration, but for numerical integration we have to use the additional evalf command

**Some inbuilt
functions in Maple
being used
for integration**

Numerical Integration

Calling Sequences

evalf(Int(f, x=a..b))

evalf(Int(f, a..b))

evalf(Int(f, x=a..b, opts))

evalf(Int(f, a..b, opts))

evalf(int(f, x=a..b))

We Define Parameters

f - algebraic expression or
procedure; integrand

x - name; variable of
integration

a,b - endpoints of the interval
of integration

opts - (optional) name or
equation of the form

option=name; options

Description

In the case of a definite integral, which is returned unevaluated, numerical integration can be invoked by applying `evalf` to the unevaluated integral. To invoke numerical integration without first invoking symbolic integration, use the inert function `Int` as in: `evalf( Int(f, x=a..b) )`.

If the integrand f is specified as a procedure or a Maple operator, then the second argument must be the range $a..b$ and not an equation. (i.e., a variable of integration must not be specified.)

```
>evalf(Int( exp(-x^3), x = 0..1 ));  
      .8075111821
```

```
> evalf(Int( 1/(1+x^2), x = 0..infinity ));  
      1.570796327
```

```
> evalf(Int( sin(x)*ln(x), x = 0..1 ));  
      -0.2398117420
```

>alg041(); This is Simpson's Method.

`Input the function  $F(x)$  in terms of  $x$ `

`For example:

> sin (x)

**`Input lower limit of integration and
upper limit of integration`**

`separated by a blank`

> 0 3.14159265359

`Input an even positive integer  $N$ .`

> 20

**The integral of F from
0.000000000
to
3.14159265
is 2.000000678**

alg041(); This is Simpson's Method.

`Input the function F(x) in terms of x`

> x^2

**`Input lower limit of integration and
upper limit of integration separated
by a blank`**

>0 2

Input an even positive integer N

>20

**The integral of F from
0.000000000
to
2.000000000
is 2.666666667**

> alg041();

This is Simpson's Method.

Input the function $F(x)$ in terms of x , for example: $\cos(x)$

> $\exp(x - x^2/2)$

Input lower limit of integration and upper limit of integration separated by a blank

> 0 3.14159265359

Input an even positive integer N .

> 20

The integral of F from

0.00000000

to

3.14159265

is 3.41046542

> alg044();

**This is Simpson's Method for
double integrals.**

**Input the functions $F(X,Y)$, $C(X)$,
and $D(X)$ in terms of x and y
separated by a space.**

For example: $\cos(x+y)$ x^3 x

> $\exp(y/x)$ x^3 x^2

Input lower limit of integration and upper limit of integration separated by a blank

> 0.1 0.5

Input two even positive integers N, M ; there will be N subintervals for outer integral and M subintervals for inner integral - separate with blank

> 10 10

**The double integral of
F from**

0.100 to 0.500

Is .03330546

obtained with

N := 10 and M := 10

> alg045();

`This is Gaussian Quadrature for double integrals.`

`Input the function  $F(x,y)$  in terms of  $x$  and  $y$ `

`For example:  $\sqrt{x^2+y^2}$ `

> exp (y/x)

Input the functions $C(x)$, and $D(x)$ in terms of x separated by a space

For example: cos (x) sin (x)

> x^3 x^2

Input lower limit of integration and upper limit of integration separated by a blank space

>0.1 0.5

Input two integers $M > 1$ and $N > 1$. This implementation of Gaussian quadrature requires both to be less than or equal to 5.

M is used for the outer integral and N for the inner integral - separated by a space.

> 5 5

**The double integral of F
from**

0.1000 to 0.5000

is 3.3305566120e-02

Or 0.03305566120

obtained with

M = 5 and N = 5

Numerical Analysis Lecture 43

Numerical Analysis Lecture 44

Solution of Non-Linear Equations

Bisection Method

Regula-Falsi Method

Method of iteration

Newton - Raphson Method

Muller's Method

Graeffe's Root Squaring

Method

Newton - Raphson Method

An approximation to the root is given by

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

Better and successive approximations $x_2, x_3, \dots, x_n$ to the root are obtained from

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

N-R Formula

Newton's 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 $i \leq N0$ do Steps 3-6

Step 3

*Set $p = p_0 - f(p_0) / f'(p_0)$
(compute p_i).*

Step 4

If $Abs(p - p_0) < TOL$

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

Example

Using Maple to solve a
non-linear equation.

$$\cos(x) - x = 0$$

Solution

The Maple command will be as follows,

```
Fsolve ( cos (x) -x);
```


➤ **alg023();**

➤ **This is Newton's Method**

Input the function $F(x)$ in terms of x

For example:

> $\cos(x)-x$

Input initial approximation

> 0.7853981635

Input tolerance

> 0.00005

**Input maximum number of
iterations - no decimal point
> 25**

Select output destination

1. Screen

2. Text file

Enter 1 or 2

> 1

Select amount of output

1. Answer only

2. All intermediate

approximations

Enter 1 or 2

> 2

Newton's Method

I	P	F(P)
1	0.739536134	-7.5487470e-04
2	0.739085178	-7.5100000e-08
3	0.739085133	0.0000000e-01

Approximate solution = 0.73908513

with $F(P) = 0.000000000000$

Number of iterations = 3

Tolerance = 5.00000000000e-05

Another Example

> alg023();

**Input the function $F(x)$ in
terms of x ,**

> $\sin(x)-1$

Input initial approximation

> 0.17853

Input tolerance

> 0.00005

**Input maximum number of
iterations – no decimal point
> 25**

Select output destination

1. Screen

2. Text file

Enter 1 or 2

>2

Select amount of output

1. Answer only

**2. All intermediate
approximations**

Enter 1 or 2

> 2

Newton's Method

I	P	F(P)
1	1.01422964e+00	-1.5092616e-01
2	1.29992628e+00	-3.6461537e-02
3	1.43619550e+00	-9.0450225e-03
4	1.50359771e+00	-2.2569777e-03
5	1.53720967e+00	-5.6397880e-04
6	1.55400458e+00	-1.4097820e-04
7	1.56240065e+00	-3.5243500e-05

More...

8	1.56659852e+00	-8.81080000e-06
9	1.56869743e+00	-2.20270000e-06
10	1.56974688e+00	-5.50700000e-07
11	1.57027163e+00	-1.37700000e-07
12	1.57053407e+00	-3.44000000e-08
13	1.57066524e+00	-8.60000000e-09
14	1.57073085e+00	-2.10000000e-09
15	1.57076292e+00	-6.00000000e-10

Approximate solution

= 1.57076292

with

$F(P) = 6.000000000000e-10$

Number of iterations = 15

Tolerance =

$5.000000000000e-05$

Bisection Method

> alg021();

This is the Bisection Method.

Input the function $F(x)$ in
terms of x

For example:

> $x^3 + 4x^2 - 10$

**Input endpoints $A < B$
separated by blank**

> 1 2

Input tolerance

> 0.0005

**Input maximum number of
iterations - no decimal point**

> 25

Select output destination

1. Screen ,

2. Text file

Enter 1 or 2

> 1

Select amount of output

1. Answer only

2. All intermediate approximations

Enter 1 or 2

> 2

Bisection Method

I	P	F(P)
1	1.500000000e+00	2.37500000e+00
2	1.250000000e+00	-1.7968750e+00
3	1.375000000e+00	1.6210938e-01

4	1.31250000e+00	-8.4838867e-01
5	1.34375000e+00	-3.5098267e-01
6	1.35937500e+00	-9.6408842e-02
7	1.36718750e+00	3.2355780e-02
8	1.36328125e+00	-3.2149969e-02
9	1.36523438e+00	7.2030000e-05
10	1.36425781e+00	-1.6046697e-02
11	1.36474609e+00	-7.9892590e-03

Approximate solution $P = 1.36474609$

with $F(P) = -.00798926$

Number of iterations = 11

Tolerance = $5.000000000e-04$

**alg021(); Another example of the
Bisection Method.**

**Input the function $F(x)$ in terms
of x ,**

> cos(x)

**Input endpoints $A < B$ separated
by blank**

> 1 2

Input tolerance

> 0.0005

**Input maximum number of iterations -
no decimal point**

> 25

Select output destination

1. Screen , 2. Text file

Enter 1 or 2

> 1

Select amount of output

1. Answer only

2. All intermediate approximations

Enter 1 or 2

> 2

Bisection Method

1	P	F(P)
1	1.500000000e+00	7.0737202e-02
2	1.750000000e+00	-1.7824606e-01
3	1.625000000e+00	-5.4177135e-02
4	1.562500000e+00	8.2962316e-03
5	1.593750000e+00	-2.2951658e-02

6	1.57812500e+00	-7.3286076e-03
7	1.57031250e+00	4.8382678e-04
8	1.57421875e+00	-3.4224165e-03
9	1.57226563e+00	-1.4692977e-03
10	1.57128906e+00	-4.9273519e-04
11	1.57080078e+00	-4.4542051e-06

Approximate solution $P = 1.57080078$

with $F(P) = -.000000445$

Number of iterations = 11

Tolerance = 5.000000000e-04

➤ **alg025(); This is the Method of False Position**

Input the function $F(x)$ in terms of x

> $\cos(x)-x$

Input endpoints $P0 < P1$ separated by a blank space

➤ 0.5 0.7853981635

Input tolerance

>0.0005

**Input maximum number of
iterations - no decimal point
> 25**

Select output destination

- 1. Screen**
- 2. Text file**

Enter 1 or 2

>1

Select amount of output

1. Answer only

**2. All intermediate
approximations**

Enter 1 or 2

> 2

METHOD OF FALSE POSITION

I	P	F(P)
2	7.36384139e-01	4.51771860e-03
3	7.39058139e-01	4.51772000e-05
4	7.39084864e-01	4.50900000e-07

Approximate solution $P = .73908486$

with $F(P) = .00000045$

Number of iterations = 4

Tolerance = $5.00000000e-04$

System of Linear Equations

Gaussian Elimination

Gauss-Jordan

Elimination

Crout's Reduction

Jacobi's

Gauss- Seidal Iteration

Relaxation

Matrix Inversion

> alg061();

This is Gaussian Elimination to solve a linear system.

The array will be input from a text file in the order:

**$A(1,1), A(1,2), \dots, A(1,N+1), A(2,1),$
 $A(2,2), \dots, A(2,N+1), \dots, A(N,1), A(N,2), \dots,$
 $A(N,N+1)$**

Place as many entries as desired on each line, but separate entries with at least one blank.

Has the input file been created? - enter Y or N.

> y

**Input the file name in the form - drive:\
name.ext**

for example: A:\DATA.DTA

> d:\maple00\dta\alg061.dta

Input the number of equations - an integer.

> 4

Choice of output method:

1. Output to screen

2. Output to text file

Please enter 1 or 2.

> 1

GAUSSIAN ELIMINATION

The reduced system - output by rows:

1.00000000	-1.00000000	2.00000000	-1.00000000	-8.00000000
0.00000000	2.00000000	-1.00000000	1.00000000	6.00000000
0.00000000	0.00000000	-1.00000000	-1.00000000	-4.00000000
0.00000000	0.00000000	0.00000000	2.00000000	4.00000000

Has solution vector:

-7.00000000	3.00000000	2.00000000	2.00000000

with 1 row interchange (s)

> alg071();

This is the Jacobi Method for Linear Systems.

The array will be input from a text file in the order

$A(1,1), A(1,2), \dots, A(1,n+1), A(2,1), A(2,2), \dots,$

$A(2,n+1), \dots, A(n,1), A(n,2), \dots, A(n,n+1)$

Place as many entries as desired on each line, but separate entries with at least one blank.

The initial approximation should follow in the same format has the input file been created? - enter Y or N.

> y

Input the file name in the form - drive:\name.ext

for example: A:\DATA.DTA

> d:\maple00\alg071.dta

Input the number of equations - an integer.

> 4

Input the tolerance.

> 0.001

Input maximum number of iterations.

> 15

Choice of output method:

- 1. Output to screen**
- 2. Output to text file**

Please enter 1 or 2.

> 1

JACOBI ITERATIVE METHOD FOR LINEAR SYSTEMS

The solution vector is :

1.00011860 1.99976795

-.99982814 0.99978598

using 10 iterations

with Tolerance 1.000000000000e-03

Numerical Analysis Lecture 44

Numerical Analysis Lecture 45

Summing up

Non-Linear Equations

Bisection Method (Bolzano)

Regula-Falsi Method

Method of iteration

Newton - Raphson Method

Muller's Method

Graeffe's Root Squaring

Method

In the method of ***False Position***, the first approximation to the root of $f(x) = 0$ is given by

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n) \quad (2.2)$$

Here $f(x_{n-1})$ and $f(x_{n+1})$ are of opposite sign. Successive approximations to the root of $f(x) = 0$ is given by Eq. (2.2).

METHOD OF ITERATION
can be applied to find a real root of the equation $f(x) = 0$ by rewriting the same in the form, $x = \phi(x)$

$$\left. \begin{array}{l} x_1 = \phi(x_0) \\ x_2 = \phi(x_1) \\ \vdots \\ x_{n+1} = \phi(x_n) \end{array} \right\}$$

**In Newton – Raphson Method
successive approximations**

**$x_2, x_3, \dots, x_n$ to the root are
obtained from**

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

N-R Formula

Secant Method

$$x_{n+1} = \frac{x_{n-1}f(x_n) - x_n f(x_{n-1})}{f(x_n) - f(x_{n-1})}, n = 1, 2, 3, \dots$$

This sequence converges to the root 'b' of $f(x) = 0$ i.e. $f(b) = 0$.

**The Secant method
converges faster
than linear and
slower than
Newton's quadratic.**

In Muller's Method we can get a better approximation to the root, by using

$$x_{i+1} = x_i + h_i \lambda$$

$$\lambda = \frac{-2f_i\delta_i}{g_i \pm [g_i^2 - 4f_i\delta_i\lambda_i(f_{i-2}\lambda_i - f_{i-1}\delta_i + f_i)]^{1/2}}$$

Where we defined

$$\lambda = \frac{h}{h_i} = \frac{x - x_i}{x_i - x_{i-1}}$$

$$\lambda_i = \frac{h_i}{h_{i-1}}$$

$$\delta_i = 1 + \lambda_i$$

Systems of Linear Equations

Gaussian Elimination

Gauss-Jordan

Elimination

Crout's Reduction

Jacobi's

Gauss- Seidal Iteration

Relaxation

Matrix Inversion

In Gaussian Elimination method, the solution to the system of equations is obtained in two stages.

i) the given system of equations is reduced to an equivalent upper triangular form using elementary transformations

ii) the upper triangular system is solved using back substitution procedure

Gauss-Jordan method is a variation of Gaussian method. In this method, the elements above and below the diagonal are simultaneously made zero

In *Crout's Reduction* Method the coefficient matrix $[A]$ of the system of equations is decomposed into the product of two matrices $[L]$ and $[U]$, where $[L]$ is a lower-triangular matrix and $[U]$ is an upper-triangular matrix with 1's on its main diagonal.

**For the purpose of illustration,
consider a general matrix in the
form**

$$\begin{bmatrix} L \end{bmatrix} \begin{bmatrix} U \end{bmatrix} = \begin{bmatrix} A \end{bmatrix}$$

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

Jacobi's Method is an iterative method, where initial approximate solution to a given system of equations is assumed and is improved towards the exact solution in an iterative way.

**In Jacobi's method,
the $(r + 1)$ *th*
approximation to the
above system is given
by Equations**

$$\left. \begin{aligned}
 x_1^{(r+1)} &= \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}} x_2^{(r)} - \dots - \frac{a_{1n}}{a_{11}} x_n^{(r)} \\
 x_2^{(r+1)} &= \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}} x_1^{(r)} - \dots - \frac{a_{2n}}{a_{22}} x_n^{(r)} \\
 \vdots & \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\
 x_n^{(r+1)} &= \frac{b_n}{a_{nn}} - \frac{a_{n1}}{a_{nn}} x_1^{(r)} - \dots - \frac{a_{n(n-1)}}{a_{nn}} x_{n-1}^{(r)}
 \end{aligned} \right\}$$

**Here we can observe
that no element of $x_i^{(r+1)}$
replaces $x_i^{(r)}$ entirely
for the next cycle of
computation.**

**In Gauss-Seidel method,
the corresponding
elements of $x_i^{(r+1)}$
replaces those of $x_i^{(r)}$ as
soon as they become
available. It is also called
method of Successive
Displacement.**

The Relaxation Method is also an iterative method and is due to Southwell.

$$dx_i = \frac{R_i}{a_{ii}}$$

**Eigen Value
Problems
Power Method
Jacobi's Method**

In Power Method the result looks like

$$u^{(k)} = [A]v^{(k-1)} = q_k v^{(k)}$$

Here, q_k is the desired largest eigen value and $v^{(k)}$ is the corresponding eigenvector.

Interpolation

Finite Difference Operators
Newton's Forward Difference
Interpolation Formula
Newton's Backward Difference
Interpolation Formula
Lagrange's Interpolation
Formula
Divided Differences
Interpolation in Two Dimensions
Cubic Spline Interpolation

Finite Difference Operators

Forward Differences

Backward Differences

Central Difference

$$\Delta^r y_i = \Delta^{r-1} y_{i+1} - \Delta^{r-1} y_i$$

$$\nabla^k y_i = \nabla^{k-1} y_i - \nabla^{k-1} y_{i-1},$$

$$i = n, (n-1), \dots, k$$

$$\delta^n y_i = \delta^{n-1} y_{i+(1/2)} - \delta^{n-1} y_{i-(1/2)}$$

Thus

$$\Delta y_x = y_{x+h} - y_x = f(x+h) - f(x)$$

$$\Delta^2 y_x = \Delta y_{x+h} - \Delta y_x$$

Similarly

$$\nabla y_x = y_x - y_{x-h} = f(x) - f(x-h)$$

$$\delta y_x = y_{x+(h/2)} - y_{x-(h/2)} = f\left(x + \frac{h}{2}\right) - f\left(x - \frac{h}{2}\right)$$

Shift operator, E

$$E f(x) = f(x + h)$$

$$E^n f(x) = f(x + nh)$$

$$E^n y_x = y_{x+nh}$$

**The inverse operator E^{-1}
is defined as**

$$E^{-1} f(x) = f(x - h)$$

Similarly,

$$E^{-n} f(x) = f(x - nh)$$

Average Operator, μ

$$\begin{aligned}\mu f(x) &= \frac{1}{2} \left[f\left(x + \frac{h}{2}\right) + f\left(x - \frac{h}{2}\right) \right] \\ &= \frac{1}{2} \left[y_{x+(h/2)} + y_{x-(h/2)} \right]\end{aligned}$$

Differential Operator, D

$$Df(x) = \frac{d}{dx} f(x) = f'(x)$$

$$D^2 f(x) = \frac{d^2}{dx^2} f(x) = f''(x)$$

Important Results

$$\Delta = E - 1$$

$$\nabla = 1 - E^{-1} = \frac{E - 1}{E}$$

$$\delta = E^{1/2} - E^{-1/2}$$

$$hD = \log E$$

$$\mu = \frac{1}{2} (E^{1/2} + E^{-1/2})$$

The Newton's forward difference formula for interpolation, which gives the value of $f(x_0 + ph)$ in terms of $f(x_0)$ and its leading differences.

$$\begin{aligned} f(x_0 + ph) = & f(x_0) + p\Delta f(x_0) \\ & + \frac{p(p-1)}{2!} \Delta^2 f(x_0) + \frac{p(p-1)(p-2)}{3!} \Delta^3 f(x_0) \\ & + \dots + \frac{p(p-1)\dots(p-n+1)}{n!} \Delta^n f(x_0) + \text{Error} \end{aligned}$$

An alternate expression is

$$\begin{aligned} y_x = & y_0 + p\Delta y_0 + \frac{p(p-1)}{2!} \Delta^2 y_0 \\ & + \frac{p(p-1)(p-2)}{3!} \Delta^3 y_0 + \dots \\ & + \frac{p(p-1)\cdots(p-n+1)}{n!} \Delta^n y_0 + \text{Error} \end{aligned}$$

Newton's Backward difference formula is,

$$\begin{aligned} f(x_n + ph) &= f(x_n) + p\nabla f(x_n) \\ &+ \frac{p(p+1)}{2!} \nabla^2 f(x_n) \\ &+ \frac{p(p+1)(p+2)}{3!} \nabla^3 f(x_n) + \dots \\ &+ \frac{p(p+1)(p+2)\cdots(p+n-1)}{n!} \nabla^n f(x_n) + \text{Error} \end{aligned}$$

**Alternatively, this formula
can also be written as**

$$\begin{aligned} y_x = & y_n + p\nabla y_n + \frac{p(p+1)}{2!} \nabla^2 y_n \\ & + \frac{p(p+1)(p+2)}{3!} \nabla^3 y_n + \dots \\ & + \frac{p(p+1)(p+2)\cdots(p+n-1)}{n!} \nabla^n y_n + \text{Error} \end{aligned}$$

Here

$$p = \frac{x - x_n}{h}$$

The Lagrange's formula for interpolation

$$\begin{aligned} y = f(x) = & \frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)} y_0 + \frac{(x-x_0)(x-x_2)\cdots(x-x_n)}{(x_1-x_0)(x_1-x_2)\cdots(x_1-x_n)} y_1 + \cdots \\ & + \frac{(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)} y_i + \cdots \\ & + \frac{(x-x_0)(x-x_1)(x-x_2)\cdots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)(x_n-x_2)\cdots(x_n-x_{n-1})} y_n \end{aligned}$$

Newton's divided difference interpolation formula can be written as

$$\begin{aligned} y = f(x) = & y_0 + (x - x_0)y[x_0, x_1] \\ & + (x - x_0)(x - x_1)y[x_0, x_1, x_2] \\ & + \dots + \\ & (x - x_0)(x - x_1) \dots (x - x_{n-1})y[x_0, x_1, \dots, x_n] \end{aligned}$$

Where the first order divided difference is defined as

$$y[x_0, x_1] = \frac{y_1 - y_0}{x_1 - x_0}$$

Numerical Differentiation and Integration

We expressed D in terms of Δ :

$$D = \frac{1}{h} \left(\Delta - \frac{\Delta^2}{2} + \frac{\Delta^3}{3} - \frac{\Delta^4}{4} + \frac{\Delta^5}{5} - \dots \right)$$

Using backward difference operator ∇ , we have

$$hD = -\log(1 - \nabla).$$

On expansion, we have

$$D = \frac{1}{h} \left(\nabla - \frac{\nabla^2}{2} + \frac{\nabla^3}{3} - \frac{\nabla^4}{4} + \dots \right)$$

Using Central difference Operator

$$D = \frac{1}{h} \left(\delta - \frac{1}{24} \delta^3 + \frac{3}{640} \delta^5 - \dots \right)$$

**Differentiation Using Interpolation
Richardson's Extrapolation**

$$\frac{d}{dx} \prod_{i=0}^{n-1} (x - x_i)$$

$$= \sum_{i=0}^{n-1} \frac{(x - x_0)(x - x_1) \cdots (x - x_n)}{x - x_i}$$

Thus, $y'(x)$ is approximated by $P'_n(x)$ which is given by

$$P'_n(x) = y[x_0, x_1] + [(x - x_1) + (x - x_0)]y[x_0, x_1, x_2] + \cdots +$$

$$\sum_{i=0}^{n-1} \frac{(x-x_0)(x-x_1)\cdots(x-x_{n-1})}{x-x_i}$$

$$y[x_0, x_1, \dots, x_n]$$

$$F_m \left(\frac{h}{2^m} \right) \\ = \frac{4^m F_{m-1} \left(\frac{h}{2^m} \right) - F_{m-1} \left(\frac{h}{2^{m-1}} \right)}{4^m - 1},$$

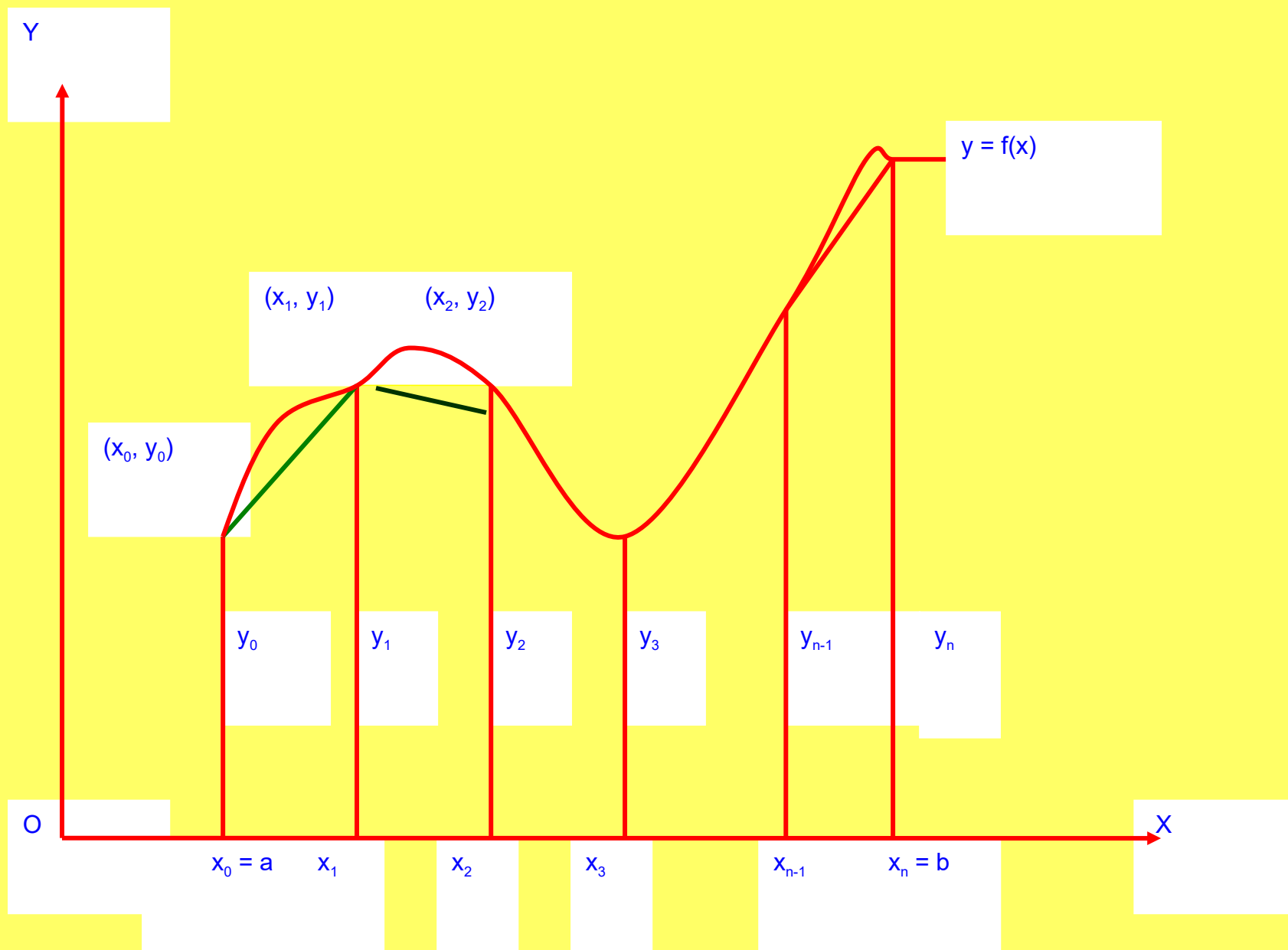
$$m = 1, 2, 3, \dots$$

Basic Issues in Integration

What does an integral represent?

$$\int_a^b f(x) dx = \textit{AREA}$$

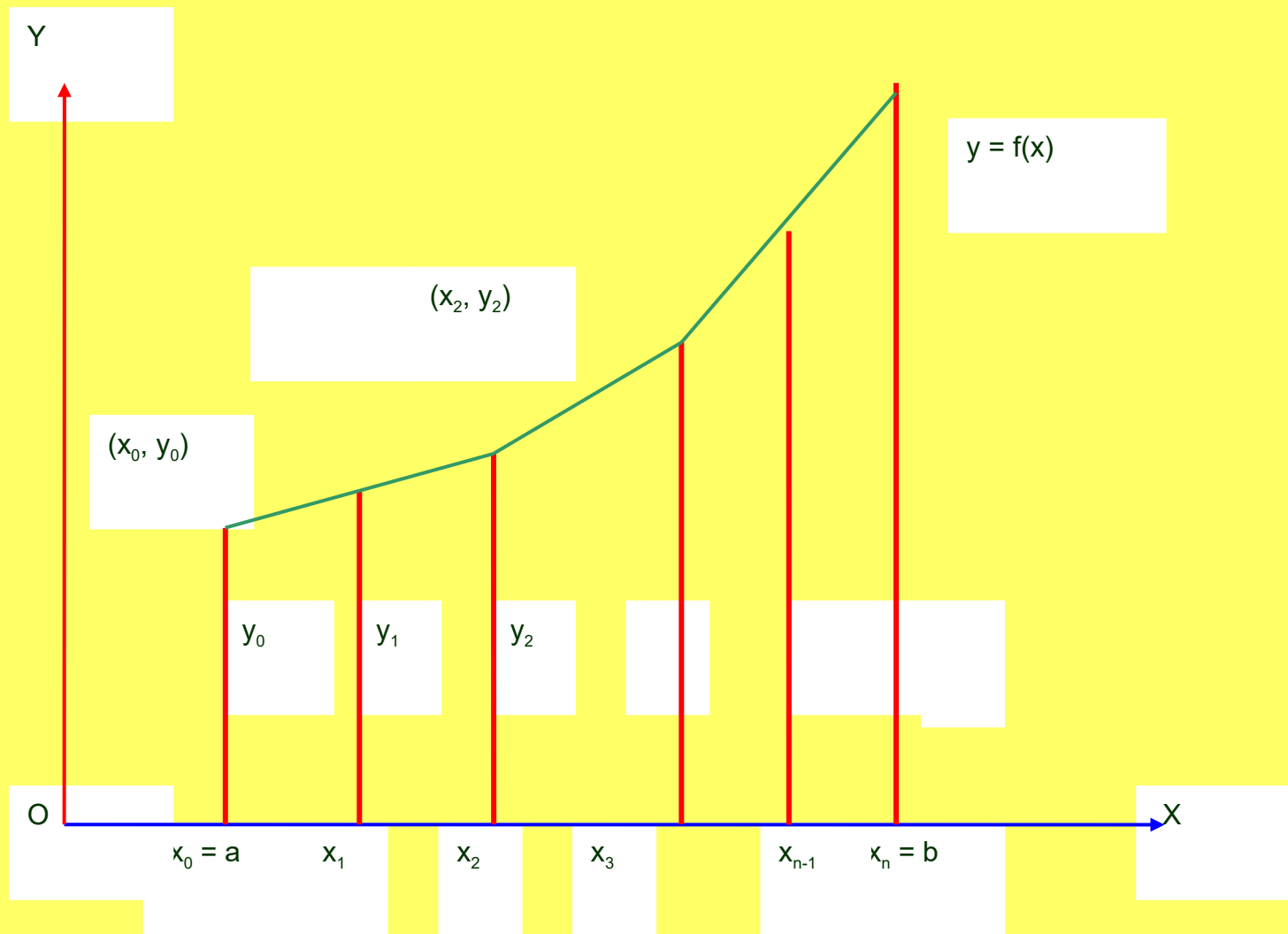
$$\int_c^d \int_a^b g(x, y) dx dy = \textit{VOLUME}$$



$$\int_{x_0}^{x_3} f(x) dx$$

$$= \frac{3}{8} h (y_0 + 3y_1 + 3y_2 + y_3)$$

$$- \frac{3}{80} h^5 y^{(iv)}(\xi)$$



TRAPEZOIDAL RULE

$$\int_{x_0}^{x_n} f(x) dx$$

$$= \frac{h}{2} (y_0 + 2y_1 + 2y_2$$

$$+ \cdots + 2y_{n-1} + y_n) + E_n$$

DOUBLE INTEGRATION

We described procedure to evaluate numerically a double integral of the form

$$I = \int \left[\int (x, y) dx \right] dy$$

Differential Equations

Taylor Series

Euler Method

Runge-Kutta Method

Predictor Corrector

Method

In Taylor's series we expanded $y(t)$ by Taylor's series about the point $t = t_0$ and obtain

$$y(t) = y(t_0) + (t - t_0)y'(t_0) + \frac{(t - t_0)^2}{2!}y''(t_0)$$

$$+ \frac{(t - t_0)^3}{3!}y'''(t_0) + \frac{(t - t_0)^4}{4!}y^{IV}(t_0) + \dots$$

In Euler Method we obtained the solution of the differential equation in the form of a recurrence relation

$$y_{m+1} = y_m + hf(t_m, y_m)$$

We derived the recurrence relation

$$y_{m+1} = y_m + h \left[\frac{f(t_m, y_m) + f(t_{m+1}, y_{m+1}^{(1)})}{2} \right]$$

Which is the *modified Euler's method*.

The fourth-order R-K method was described as

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

where

$$k_1 = hf(t_n, y_n)$$

$$k_2 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$

$$k_3 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right)$$

$$k_4 = hf(t_n + h, y_n + k_3)$$

In general, Milne's predictor-corrector pair can be written as

$$\left. \begin{aligned} P : y_{n+1} &= y_{n-3} + \frac{4h}{3}(2y'_{n-2} - y'_{n-1} + 2y'_n) \\ C : y_{n+1} &= y_{n-1} + \frac{h}{3}(y'_{n-1} + 4y'_n + y'_{n+1}) \end{aligned} \right\}$$

This is known as Adam's predictor formula.

$$y_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) + \frac{251}{720}h\nabla^4 f_n$$

Alternatively, it can be written as

$$y_{n+1} = y_n + \frac{h}{24}[55y'_n - 59y'_{n-1} + 37y'_{n-2} - 9y'_{n-3}] + \frac{251}{720}h\nabla^4 y'_n$$

Numerical Analysis Lecture 45

www.vuCtn.com

Good Luck

Regards : Zarva Chaudhary

Admin:

* Zarva Chaudhary *

* Chaudhary Moazzam *

* Laiba Maki *