* Curve fitting ÷

The process of determining the constant, involved in a curve with the help of given data point by least square method is known as curve fitting.

* Least square method :

It is a technique that minimize the sum of square of errors.

* Types of curve ÷>

1. Linear curve ; $y = a + bx$
2. Quadratic curve ; $y = a + bx + cx^2$
3. Power Curve ; $y = ax^b$
4. Exponential curve ; $y = ae^{bx}$, ab^x

Scanned by CamScanner

~~when~~ $2\Sigma(y-a-bx)\times(-1)=0$

or, $-2\Sigma(y-a-bx)=0$

$\because -2 \neq 0$ so

$\Sigma(y-a-bx)=0$

$\Sigma y-\Sigma a-b\Sigma x=0$

$\Sigma y-na-b\Sigma x=0$

$\therefore \boxed{\Sigma y = na+b\Sigma x}$ —(iii)

$$\frac{\partial^2 s}{\partial a^2}=\frac{\partial}{\partial a}\left(\frac{\partial s}{\partial a}\right)$$

$$=\frac{\partial}{\partial a}\left[-2\Sigma(y-a-bx)\right]$$

$$=-2\times(-1)=2>0$$

$$\therefore \frac{\partial^2 s}{\partial a^2}>0$$

When $\frac{\partial s}{\partial b}=0$

$$\frac{\partial}{\partial b}\Sigma(y-a-bx)^2=0$$

$$+2\Sigma(y-a-bx)\cdot(-x)=0$$

$$-2\Sigma(y-a-bx)\cdot x=0$$

$$-2\Sigma(yx-ax-bx^2)=0$$

$\Sigma(xy-ax-bx^2)=0$ $[\because -2\neq 0]$

$\therefore \boxed{\Sigma xy = a\Sigma x+b\Sigma x^2}$ —(iv)

$$\frac{\partial^2 s}{\partial b^2}=\frac{\partial}{\partial b}\left(\frac{\partial s}{\partial b}\right)=\frac{\partial}{\partial s}\left[-2\Sigma(xy-ax-bx^2)\right]$$

$$=-2\times(-x^2)$$

$$=x^2>0$$

$$\therefore \frac{\partial^2 s}{\partial b^2}>0$$

Eqns (iii) & (iv) are known as normal equations. Solving eqn (iii) & (iv) with the help of given data points we get the value of a & b. Now putting the value of a & b in eqn (i) we get

$y = __ + (__)\,x$

which is fitted linear curve.

Q. fit the linear curve to the following data pts

x:	1	2	3	4
y:	2	5	7	10

soln:→

Let $y = a+bx$ —i) be linear curve.

The normal equations are

$\Sigma y = na+b\Sigma x$ —ii)

$\Sigma xy = a\Sigma x+b\Sigma x^2$ —iii)

Scanned by CamScanner

x	y	x^2	xy
1	2	1	2
2	5	4	10
3	7	9	21
4	10	16	40
$\Sigma x = 10$	$\Sigma y = 24$	$\Sigma x^2 = 30$	$\Sigma xy = 73$

Using above data in eqns (ii) & (iii)

$$4a + 10b = 24 \quad \text{— (iv)}$$

$$10a + 30b = 73 \quad \text{— (v)}$$

Solving eqns iv) & v)

$$4a + 10b = 24 \quad] \times 3$$

$$\underline{10a + 30b = 73}$$

$$2a = -1$$

$$a = -\frac{1}{2} = -0.5$$

$$\therefore b = \frac{24 - 4\times(-\frac{1}{2})}{10} = 2.6$$

putting the value of a & b in eqn (1), we get

$$y = -0.5 + 2.6x$$

x	y	x^2	x^3	x^4	xy	x^2y
0	2	0	0	0	0	6
1	3	1	1	1	3	3
2	5	4	8	16	10	20
3	8	9	27	81	24	72
$\Sigma x = 6$	$\Sigma y = 18$	$\Sigma x^2 = 14$	$\Sigma x^3 = 36$	$\Sigma x^4 = 98$	$\Sigma xy = 37$	$\Sigma x^2y = 101$

from equations ii), iii) & iv)

$4a + 6b + 14c = 18$ — v)

$6a + 14b + 36c = 37$ — vi)

$14a + 36b + 98c = 101$ — vii)

Solving above eqn by Cramers method

* fitting of Exponential curve : ($y = ae^{bx}$)

Given

$y = ae^{bx}$ — i) be the exponential curve

Taking log on both sides

$\log y = \log a + bx \log e$

i.e.

$Y = A + Bx$ — ii) which is linear in x

where,

$Y = \log y$

$A = \log a$

$\Rightarrow a = \text{antilog} A$

$B = b \log e$

$\Rightarrow b = \dfrac{B}{\log e} = \dfrac{B}{0.4343}$

$X = x$

The normal equations of eqn ii)

$\Sigma Y = nA + B\Sigma X$ — iii)

$\Sigma XY = A\Sigma X + B\Sigma x^2$ — iv)

Solving eqn iii) & iv) with the help of given data points we get

$A =$

$B =$

$\therefore a = \text{antilog} A$

&

$b = \dfrac{B}{0.4343}$

Now putting the value of a & b in eqn i) $y = (_)e^{(_)x}$

which is fitted exponential curve of type $y = ae^{bx}$

* Fitting of an exponential curve $y = ab^x$

Given

$y = ab^x$ — i) be the exponential curve

Taking log on both sides

$\log y = \log a + x \log b$

i.e.

$Y = A + Bx$ — ii) which is linear

where, $Y = \log y$

$A = \log a$

$\Rightarrow a = \text{antilog} A$

$B = \log b$

$\Rightarrow b = \text{antilog} B$

& $X = x$

So normal eqns are

$\Sigma Y = nA + b\Sigma X$ — iii)

$\Sigma XY = A\Sigma X + b\Sigma X^2$ — iv)

Solving eqns iii) & iv) with the help of given data points

$A = \ldots$

$B = \ldots$

$\therefore a = \text{antilog} A$

$b = \text{antilog} B$

Now, putting the value of a & b in eqn i), we get

$y = (\ldots)(\ldots)^x$ — v)

which is fitted exponential curve of type $y = ab^x$

eg: fit the exponential curve a) $y = ae^{bx}$ b) $y = ab^x$ to the following data points

x:	1	2	3	4
y:	1	4	9	16

Soln ⇒

(a) $y = ae^{bx}$ — i)

Taking log on both sides

$\log y = \log a + bx \log e$

i.e.

$Y = A + Bx$ — ii)

where $Y = \log y$

$A = \log a \Rightarrow a = \text{antilog} A$

$B = b \log e \Rightarrow b = \frac{B}{0.4343}$

& $X = x$

The normal equations are

$\Sigma Y = nA + b\Sigma X$ — iii)

$\Sigma XY = A\Sigma X + b\Sigma X^2$ — iv)

$X = x$	$Y = \log y$	X^2	XY
1	0	1	0
2	0.602	4	1.204
3	0.954	9	2.862
4	1.204	16	4.816
$\Sigma X = 10$	$\Sigma Y = 2.760$	$\Sigma X^2 = 30$	$\Sigma XY = 8.882$

So eqns iii) & iv) becomes

$4A + 10b = 2.760$ — v)

$10A + 30b = 8.882$ — vi)

$4A + 10b = 2.760$ $\quad | \times 3$

$10A + 30b = 8.882$

$2A = -0.602$

$A = -0.301$

using A in eqn v)

$$4\times(-0.301) + 10B = 2.760$$

$$\therefore 10B = 3.964$$

$$\therefore B = 0.3964$$

$\therefore a = \text{antilog}\, A = \text{antilog}(-0.301)$

$a = 0.50$

$$b = \frac{B}{0.4343} = \frac{0.3964}{0.4343} = 0.913$$

using a & b in eqn i)

$$y = 0.50\, e^{0.913x}$$

which is fitted curve of type $y = ae^{bx}$

(b) $y = ab^x$

Scanned by CamScanner

* fitting of a power curve, $y = ax^b$

Given power curve is $y = ax^b$ — i)

Taking log on both sides

$$\log y = \log a + b \log x \quad \text{— ii)}$$

i.e. $Y = A + BX$ — ii)

where $Y = \log y$

$A = \log a \Rightarrow a = \text{antilog } A$

$B = b$

$X = \log x$

The normal equations of eqn ii) are

$$\Sigma Y = nA + B\Sigma X \quad \text{— iii)}$$

$$\Sigma XY = A\Sigma X + B\Sigma X^2 \quad \text{— iv)}$$

Solving equations iii) & iv) with the help of given points we get

$A =$

$B =$

$\therefore a = \text{antilog } A =$

$b = B$

putting the value of a and b in eqn i)

we get

$$y = (\cdots)\, x^{(\quad)} \quad \text{— (v)}$$

which is fitted power curve.

eg; fit the power curve to the following data

Points					
	x :	2	5	10	12
	y :	3	8	15	20

Soln:

Let us consider the power curve $y = ax^b$ — i)

Taking log on both sides

$$\log y = \log a + b \log x$$

Scanned by CamScanner

* INTERPOLATION :-

Let $y = f(x)$ be any function where x is input and y is output, then the process of predicting the output inside the given range of input is known as interpolation.

When the output is predicted outside the given range of input then it is called extrapolation.

Types Of Interpolation :-

There are two types of interpolation. They are-

1) Interpolation when inputs are at equal interval.
2) Interpolation when inputs are not at equal interval

Methods Of finding Interpolation when inputs are at equal interval

a) Newton's forward difference method
b) Newton's backward difference method
c) Gaus forward difference method
d) Gaus backward difference method.
e) Stirling method
f) Everett's method
g) Bessel's method.

Methods Of finding Interpolation when inputs are not at equal interval.

a) Lagrange's interpolation polynomial method
b) Newton's divided difference method.

Scanned by CamScanner

* Newton's forward Difference Method :-

This method is used when inputs are at equal interval and the interpolation is required in the begining of the table.
This method uses forward difference table.

forward Difference Table:

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
x_0	y_0	$y_1 - y_0 = \Delta y_0$	$\Delta y_1 - \Delta y_0 = \Delta^2 y_0$	$\Delta^2 y_1 - \Delta^2 y_0 = \Delta^3 y_0$	$\Delta^3 y_1 - \Delta^3 y_0 = \Delta^4 y_0$
x_1	y_1	$y_2 - y_1 = \Delta y_1$	$\Delta y_2 - \Delta y_1 = \Delta^2 y_1$	$\Delta^2 y_2 - \Delta^2 y_1 = \Delta^3 y_1$	
x_2	y_2	$y_3 - y_2 = \Delta y_2$	$\Delta y_3 - \Delta y_2 = \Delta^2 y_2$		
x_3	y_3	$y_4 - y_3 = \Delta y_3$			
x_4	y_4				

formula :-

$$\boxed{y = y_0 + u\Delta y_0 + \frac{u(u-1)}{2!}\Delta^2 y_0 + \frac{u(u-1)(u-2)}{3!}\Delta^3 y_0 + \frac{u(u-1)(u-2)(u-3)}{4!}\Delta^4 y_0 + \cdots}$$

where $u = \frac{x - x_0}{h}$

where
x = interpolating point
x_0 = initial value of x
h = common difference

e.g.

Years(x)	1990	1995	2000	2005
pop (y)	5	8	10	15

Estimate the population for year 2003

Solution:

x	y	∇y	$\nabla^2 y$	$\nabla^3 y$
1990	5			
		$3 = \nabla y_1$		
1995	8		$-1 = \nabla^2 y_2$	
		$2 = \nabla y_2$		$4 = \nabla^3 y_3$
2000	10		$3 = \nabla^2 y_3$	
		$5 = \nabla y_3$		
2005 $= x_n$	$15 = y_3$			

We have

$$y = y_n + u\nabla y_n + \frac{u(u+1)}{2!}\nabla^2 y_n + \frac{u(u+1)(u+2)}{3!}\nabla^3 y_n + \cdots$$

Here $y_n = y_3$ so,

$$y = y_3 + u\nabla y_3 + \frac{u(u+1)}{2!}\nabla^2 y_3 + \frac{u(u+1)(u+2)}{3!}\nabla^3 y_3 + \cdots$$

where

$$u = \frac{x - x_n}{h}$$

$$= \frac{x - x_3}{h} = \frac{20003 - 20005}{5} = -0.4$$

$$\therefore y = 15 + (-0.4)\times 5 + \frac{(-0.4)(-0.4+1)}{2}\times 3 + \frac{(-0.4)(-0.4+1)(-0.4+2)}{3!}\times 4$$

$$=$$

* Lagrange Interpolation Polynomial Method :-

Let $y = f(x)$ be any function where x can take values $x_0, x_1, x_2, \ldots, x_n$ and y can take values $y_0, y_1, y_2, \ldots, y_n$. Then Lagrange interpolation polynomial is given by

$$f(x) = \frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)} y_0 + \frac{(x-x_0)(x-x_2)\cdots(x-x_n)}{(x_1-x_0)(x_1-x_2)\cdots(x_1-x_n)} y_1$$

$$+ \cdots + \frac{(x-x_0)(x-x_1)\cdots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\cdots(x_n-x_{n-1})} y_n \quad \text{— (i)}$$

which is Lagrange interpolation polynomial

Q. find the cubic polynomial and hence evaluate the value of $y(3)$ from the following table.

x :	2	5	10	12
y :	3	8	12	15

Solution :-

x	y
$x_0 = 2$	$y_0 = 3$
$x_1 = 5$	$y_1 = 8$
$x_2 = 10$	$y_2 = 12$
$x_3 = 12$	$y_3 = 15$

We have

$$y = \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} y_0 + \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} y_1 +$$

$$\frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} y_2 + \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} y_3$$

Scanned by CamScanner

* Numerical Differentiation:

⟨A⟩ Let us consider Newton's forward Difference formula:

$$y = y_0 + u\Delta y_0 + \frac{u(u-1)}{2!}\Delta^2 y_0 + \frac{u(u-1)(u-2)}{3!}\Delta^3 y_0 + \cdots \quad \text{— i)}$$

where $u = \frac{x - x_0}{h}$ — ii)

$$\therefore \frac{du}{dx} = \frac{1}{h}$$

$$\frac{dy}{dx} = \frac{dy}{du} \times \frac{du}{dx}$$ [∵ y is the function of u in eqn i)]

$$\frac{dy}{dx} = \frac{1}{h}\left[\Delta y_0 + \frac{1}{2}(2u-1)\Delta^2 y_0 + \frac{1}{6}(3u^2 - 6u + 2)\Delta^3 y_0 + \cdots\right] \quad \text{— iii)}$$

$$\frac{d^2y}{dx^2} = \frac{d}{du}\left(\frac{dy}{dx}\right) \times \frac{du}{dx}$$

$$= \frac{1}{h^2}\left[\Delta^2 y_0 + \frac{1}{6}(6u-6)\Delta^3 y_0 + \cdots\right]$$

$$\therefore \frac{d^2y}{dx^2} = \frac{1}{h^2}\left[\Delta^2 y_0 + (u-1)\Delta^3 y_0 + \cdots\right]$$

⟨B⟩ Newton's Backward Difference formula for Derivative:

We have

$$y = y_n + u\nabla y_n + \frac{u(u+1)}{2!}\nabla^2 y_n + \frac{u(u+1)(u+2)}{3!}\nabla^3 y_n + \cdots \quad \text{— i)}$$

where $u = \frac{x - x_n}{h}$ — ii)

$$\therefore \frac{du}{dx} = \frac{1}{h}$$

$$\frac{dy}{dx} = \frac{dy}{du} \times \frac{du}{dx}$$

$$\therefore \frac{dy}{dx} = \frac{1}{h}\left[\nabla y_n + \frac{1}{2}(2u+1)\nabla^2 y_n + \frac{1}{6}(3u^2 + 6u + 2)\nabla^3 y_n + \cdots\right] \quad \text{— iii)}$$

$$\frac{d^2y}{dx^2} = \frac{d}{du}\left(\frac{dy}{dx}\right) \times \frac{du}{dx}$$

$$\therefore \frac{d^2y}{dx^2} = \frac{1}{h^2}\left[\nabla^2 y_n + \frac{1}{6}(6u+6)\nabla^3 y_n + \cdots\right] \quad \text{— iv)}$$

e.g:

Year	1990	1995	2000	2005	2010
population	2	5	10	12	15

find the first derivative at the point 1995

Soln ⇒

$\frac{dy}{dx}$ $\frac{d^2y}{dx^2}$

We have

$$y = y_0 + u\Delta y_0 + \frac{u(u-1)}{2!}\Delta^2 y_0 + \frac{u(u-1)(u-2)}{3!}\Delta^3 y_0 + \frac{u(u-1)(u-2)(u-3)}{4!}\Delta^4 y_0$$

where $u = \frac{x - x_0}{h}$, $\frac{du}{dx} = \frac{1}{h}$

$$\therefore \frac{dy}{dx} = \frac{dy}{du} \times \frac{du}{dx}$$

Scanned by CamScanner

$$\therefore \frac{dy}{dx} = \frac{1}{h}\left[\Delta y_0 + \frac{1}{2}(2u-1)\Delta^2 y_0 + \frac{1}{6}(3u^2-6u+2)\Delta^3 y_0 + \frac{1}{24}(4u^3-18u^2+22u-6)\Delta^4 y_0\right] \quad \text{—(1)}$$

x(year)	y(pop)	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
1990	2	$3=\Delta y_0$	$2=\Delta^2 y_0$	$-5=\Delta^3 y_0$	$9=\Delta^4 y_0$
1995	5	5	-3	4	
2000	10	2	1		
2005	12	3			
2010	15				

$$u = \frac{x-x_0}{h} = \frac{1995-1990}{5} = 1$$

$$\therefore \frac{dy}{dx} = \frac{1}{5}\left[3 + \frac{1}{2}(2\cdot1-1)\cdot2 + \frac{1}{6}(3\cdot1^2-6\cdot1+2)\cdot(-5) + \frac{1}{24}(4\cdot1^3-18\cdot1^2+22\cdot1-6)\cdot9\right]$$

$$= \frac{1}{5}\left[3+1+\frac{5}{6}+\frac{9}{4}\right]$$

$$= 1.1167$$

page: 358 (Eg 11.8)
367 (11, 12, 13, 14)

* Numerical Integration:-

Let $y = f(x)$ be any function where x can take values $x_0, x_1, x_2, \ldots, x_n$ and y can take the values $y_0, y_1, y_2, \ldots, y_n$ Then the integration of y from x_0 to x_n is given by

$$I = \int_{x_0}^{x_n} y\,dx$$

Let us consider Newton's forward difference formula for interpolation then

$$I = \int_{x_0}^{x_n} y\,dx = \int_{x_0}^{x_n} \left(y_0 + u\Delta y_0 + \frac{u(u-1)}{2!}\Delta^2 y_0 + \frac{u(u-1)(u-2)}{3!}\Delta^3 y_0 + \cdots \right) dx$$

where $u = \frac{x-x_0}{h}$

$x - x_0 = uh$

$\therefore dx = h\,du$

when $x = x_0$, $u = 0$

$x = x_n$, $u = \frac{x_n - x_0}{h}$

$= \frac{nh}{h} = n$

$\therefore \boxed{u = n}$

Rough
$x_1 - x_0 = h$
$x_2 - x_1 = h$
$x_3 - x_2 = h$
$\vdots$
$x_n - x_{n-1} = h$
Adding
$x_n - x_0 = nh$

Now,

$$\therefore I = \int_{x_0}^{x_n} y\,dx = \int_0^n \left(y_0 + u\Delta y_0 + \frac{u(u-1)}{2!}\Delta^2 y_0 + \frac{u(u-1)(u-2)}{3!}\Delta^3 y_0 + \cdots \right) h\,du$$

$$= h\left[u y_0 + \frac{u^2}{2}\Delta y_0 + \frac{1}{2}\left(\frac{u^3}{3} - \frac{u^2}{2}\right)\Delta^2 y_0 + \frac{1}{6}\left(\frac{u^4}{4} - \frac{3}{3}u^3 + \frac{2u^2}{2}\right)\Delta^3 y_0 + \cdots \right]_0^n$$

$$\therefore \int_{x_0}^{x_n} y\,dx = h\left[ny_0 + \frac{n^2}{2}\Delta y_0 + \frac{1}{2}\left(\frac{n^3}{3} - \frac{n^2}{2}\right)\Delta^2 y_0 + \frac{1}{6}\left(\frac{n^4}{4} - n^3 + n^2\right)\Delta^3 y_0 + \cdots\right] \quad \text{--- i)}$$

Equation ① is known as Newton's quadrature formula for the integration.

Trapezoidal Rule:

putting $n=1$ in equation ①

$$\int_{x_0}^{x_1} y\,dx = h\left[y_0 + \frac{1}{2}\Delta y_0\right]$$

[$\because$ integration is only from x_0 to x_1]

$$\therefore = h\left[y_0 + \frac{1}{2}(y_1 - y_0)\right]$$

$$= h\left[\frac{2y_0 + y_1 - y_0}{2}\right]$$

$$\therefore \int_{x_0}^{x_1} y\,dx = \frac{h}{2}\left[y_0 + y_1\right]$$

Similarly, $\int_{x_1}^{x_2} y\,dx = \frac{h}{2}\left[y_1 + y_2\right]$

& $\int_{x_2}^{x_3} y\,dx = \frac{h}{2}\left[y_2 + y_3\right]$

$\vdots \qquad \vdots$

$$\int_{x_{n-1}}^{x_n} y\,dx = \frac{h}{2}\left[y_{n-1} + y_n\right]$$

We know that,

$$\int_{x_0}^{x_n} y\,dx = \int_{x_0}^{x_1} y\,dx + \int_{x_1}^{x_2} y\,dx + \int_{x_2}^{x_3} y\,dx + \cdots + \int_{x_{n-1}}^{x_n} y\,dx$$

$$= \frac{h}{2}\left[y_0 + y_1 + y_1 + y_2 + y_2 + y_3 + \cdots + y_{n-1} + y_n\right]$$

$$\boxed{\int_{x_0}^{x_n} y\,dx = \frac{h}{2}\left[y_0 + y_n + 2(y_1 + y_2 + \cdots + y_{n-1})\right]} \quad \text{--- ii)}$$

which is known as trapezoidal rule for integration

Q. Evaluate $\int_0^6 x^2\,dx$ using trapezoidal Rule. Also calculate the error.

Solution:

Given $y = x^2$,

$x_0 = 0$

$x_n = 6$

$h = \frac{x_n - x_0}{n}$

$h = \frac{6-0}{6}$ [choose $n=6$]

$\therefore h = 1$

n is the no. of strips

$x_0 \to$ one strip

$x_1 \to$ 2nd strip

$x_2 \to$ 3rd strip

x_3

value of n is choosen from 5 to 10, suitable value is choosen

x	$y (= x^2)$
$x_0 = 0$	$y_0 = 0$
$x_1 = 0+1 = 1$	$y_1 = 1$
$x_2 = 1+1 = 2$	$y_2 = 4$
$x_3 = 3$	$y_3 = 9$
$x_4 = 4$	$y_4 = 16$
$x_5 = 5$	$y_5 = 25$
$x_6 = 6$	$y_6 = 36$

[$\because x_1 - x_0 = h$]

[$x_1 = h + x_0$]

$$\therefore \int_0^6 x^2\,dx = \frac{h}{2}\left[y_0+y_6+2(y_1+y_2+y_3+y_4+y_5)\right]$$

$$= \frac{1}{2}\left[0+36+2(1+4+9+16+25)\right]$$

$$= \frac{1}{2}\left[36+110\right]$$

$$= \frac{146}{2}$$

$$\therefore \int_0^6 x^2\,dx = 73$$

Actual integration $= \left[\frac{x^3}{3}\right]_0^6$

$$= \frac{6^3}{3}$$

$$= 72$$

Error $= |$Actual integration − calculated$|$

$$= |72-73| = 1$$

* Simpson's $\frac{1}{3}$ Rule for Integration:

As we know:

$$\int_{x_0}^{x_n} y\,dx = h\left[ny_0 + \frac{n^2}{2}\Delta y_0 + \frac{1}{2}\left(\frac{n^3}{3}-\frac{n^2}{2}\right)\Delta^2 y_0 + \frac{1}{6}\left(\frac{n^4}{4}-n^3+n^2\right)\Delta^3 y_0 + \cdots\right] \quad \text{— (i)}$$

putting $n=2$ in eqn (i)

$$\int_{x_0}^{x_2} y\,dx = h\left[2y_0 + 2\Delta y_0 + \frac{1}{2}\left(\frac{8}{3}-2\right)\Delta^2 y_0\right]$$

$$= h\left[2y_0 + 2(y_1-y_0) + \frac{1}{2}\times\frac{2}{3}(\Delta y_1 - \Delta y_0)\right]$$

$$= h\left[2y_0 + 2y_1 - 2y_0 + \frac{1}{3}(y_2 - y_1 - y_1 + y_0)\right]$$

$$= h\left[2y_1 + \frac{1}{3}(y_2 - 2y_1 + y_0)\right]$$

$$= \frac{h}{3}\left[6y_1 + y_2 - 2y_1 + y_0\right]$$

$$\therefore \int_{x_1}^{x_2} y\,dx = \frac{h}{3}\left[y_0 + y_2 + 4y_1\right]$$

Similarly, $\int_{x_2}^{x_4} y\,dx = \frac{h}{3}\left[y_2 + y_4 + 4y_3\right]$

$$\vdots \qquad \vdots$$

$$\int_{x_{n-2}}^{x_n} y\,dx = \frac{h}{3}\left[y_{n-2} + y_n + 4y_{n-1}\right]$$

we know that

$$\therefore \int_0^1 \frac{1}{1+x}\,dx = \frac{h}{3}\left[y_0 + y_5 + 2(y_2+y_4) + 4(y_1+y_3)\right]$$

$$= \frac{0.2}{3}\left[1+0.5+2(0.714+0.555)+4(0.833+0.625)\right]$$

$$\therefore \int_0^1 \frac{1}{1+x}\,dx = 0.658$$

Actual Integration, $\int_0^1 \frac{1}{1+x}\,dx = \left[\ln(1+x)\right]_0^1$

$$= \ln(2)$$
$$= 0.658$$

$$\therefore \text{Error} = |0.093 - 0.658| = 0.035$$

---

* Simpson's $\frac{3}{8}$ rule for Integration ⇒

We know,

$$\int_{x_0}^{x_n} y\,dx = h\left[ny_0 + \frac{n^2}{2}\Delta y_0 + \frac{1}{2}\left(\frac{n^3}{3} - \frac{n^2}{2}\right)\Delta^2 y_0 + \frac{1}{6}\left(\frac{n^4}{4} - n^3 + n^2\right)\Delta^3 y_0 + \cdots\right] \quad \text{—i)}$$

putting $n=3$ in above equation,

$$\int_{x_0}^{x_3} y\,dx = h\left[3y_0 + \frac{9}{2}\Delta y_0 + \frac{1}{2}\left(\frac{27}{3} - \frac{9}{2}\right)\Delta^2 y_0 + \frac{1}{6}\left(\frac{81}{4} - 27 + 9\right)\Delta^3 y_0\right]$$

$$= h\left[3y_0 + \frac{9}{2}(y_1 - y_0) + \frac{1}{2}\left(9 - \frac{9}{2}\right)(\Delta y_1 - \Delta y_0) + \frac{3}{8}(\Delta^2 y_1 - \Delta^2 y_0)\right]$$

$$= h\left[3y_0 + \frac{9}{2}(y_1 - y_0) + \frac{1}{2}\times\frac{9}{2}(y_2 - y_1 - y_1 + y_0) + \frac{3}{8}(\Delta y_2 - \Delta y_1 - \Delta y_1 + \Delta y_0)\right]$$

$$= \frac{h}{3}\left[3y_0 + \frac{9}{2}(y_1 - y_0) + \frac{9}{4}(y_0 + y_2 - 2y_1) + \frac{3}{8}(y_3 - y_2 - y_2 + y_1 - y_2 + y_1 + y_1 - y_0)\right]$$

8. Evaluate $\int_0^1 \frac{1}{1+x^2}\,dx$ using Simson's $\frac{3}{8}$ rule and hence find the value of π. Also calculate the error.

Soln:

$f(x) = \frac{1}{1+x^2}$, $x_0 = 0$, $x_n = 1$

$$h = \frac{x_n - x_0}{n} = \frac{1-0}{5} = 0.2$$

x	y
$x_0 = 0$	$y_0 = 1$
$x_1 = 0.2$	$y_1 = 0.962$
$x_2 = 0.4$	$y_2 = 0.862$
$x_3 = 0.6$	$y_3 = 0.735$
$x_4 = 0.8$	$y_4 = 0.611$
$x_5 = 1.0$	$y_5 = 0.5$

$$\int_{x_0}^{x_n} y\,dx = \frac{3h}{8}\left[y_0 + y_5 + 2(y_3) + 3(y_1 + y_2 + y_4)\right]$$

$$= \frac{3\times 0.2}{8}\left[1 + 0.5 + 2\times 0.735 + 3(0.962 + 0.862 + 0.611)\right]$$

$$\tan^{-1} 1 = \frac{3\times 0.2}{8}\left[1.5 + 1.47 +\right.$$

$$\frac{\pi}{4} = 0.770625$$

$$\pi = 3.0825$$

$$\text{Error} = \frac{\pi}{4} - 0.770625$$

$$= 0.014$$

Scanned by CamScanner

* Boole's Rule :-

We know

$$\int_{x_0}^{x_n} y\,dx = h\left[ny_0 + \frac{n^2}{2}\Delta y_0 + \frac{1}{2}\left(\frac{n^3}{3} - \frac{n^2}{2}\right)\Delta^2 y_0 + \frac{1}{6}\left(\frac{n^4}{4} - n^3 + n^2\right)\Delta^3 y_0 + \cdots\right] \quad \text{--- i)}$$

putting $n = 4$ in eqn i)

$$\int_{x_0}^{x_n} y\,dx = h\left[4y_0 + 8\Delta y_0 + \frac{1}{2}\left(\frac{64}{3} - 8\right)\Delta^2 y_0 + \frac{1}{6}\left(\frac{256}{4} - 64 + 16\right)\Delta^3 y_0 + \frac{1}{24}\left(\frac{1024}{5} - 384 + \frac{704}{3} - 48\right)\Delta^4 y_0\right]$$

Scanned by CamScanner

* Romberg's Integration :-

Romberg's purify the integration obtained by trapezoidal rule.

Interval	Trapezoidal Integration	Romberg's Integration	final Romberg's Integration.
h	I_1	$I_1 = I_2 + \frac{1}{3}(I_2 - I_1)$	$I = I_2 + \frac{1}{3}(I_2 - I_1)$
$\frac{1}{2}h$	I_2	$I_2 = I_3 + \frac{1}{3}(I_3 - I_2)$	
$\frac{1}{4}h$	I_3		

Q. Use Romberg's Integration to evaluate

$$\int_0^{\pi/2} \frac{\cos x}{\sqrt{1+\sin x}}\, dx$$

soln :-

Given $y = \frac{\cos x}{\sqrt{1+\sin x}}$, $x_0 = 0$, $x_n = \frac{\pi}{2}$

(i) at, $h = \frac{x_n - x_0}{n} = \frac{\pi/2 - 0}{2} = \frac{\pi}{4}$

x	$y\left(\frac{\cos x}{\sqrt{1+\sin x}}\right)$
$x_0 = 0$	$y_0 = 1$
$x_1 = \frac{\pi}{4}$	$y_1 = 0.541$
$x_2 = \frac{\pi}{2}$	$y_2 = 0$

$\therefore I_1 = \frac{h}{2}[y_0 + y_2 + 2y_1]$

$\therefore I_1 = \frac{\pi}{4}[1 + 0 + 2 \times 0.541] = 1.635$

(ii) $h = \frac{1}{2}h = \frac{1}{2} \times \frac{\pi}{4} = \frac{\pi}{8}$

Now, final Romberg's integration is calculated as,

Interval	Trapezoidal Integration	Romberg's Integration	Final Romberg's Integration
h	$I_1 = 1.635$	$I_1 = I_2 + \frac{1}{3}(I_2 - I_1)$	
$1/2\,h$	$I_2 = 0.825$	$\therefore I_1 = 0.825 + 1/3(0.825 - 1.635)$	$I = I_2 + \frac{1}{3}(I_2 - I_1)$
		$I_1 = 0.557$	$= 0.827 + 1/3(0.827 - 0.557)$
$1/4\,h$	$I_3 = 0.827$	$I_2 = 0.827 + \frac{1}{3}(0.827 - 0.825) = 0.827$	$\therefore I = 0.917$

Gaussian Integration :

It is the approximation of the form

$$I_g = \int_{-1}^{+1} f(x)\,dx$$

$$\simeq \sum_{i=1}^{n} w_i f(x_i)$$

Gaussian table page : 399 table no. 12.2

for $n=2$

$$\int_{-1}^{+1} f(x)\,dx = \sum_{i=1}^{2} w_i f(x_i)$$

$$= w_1 f(x_1) + w_2 f(x_2)$$

where $w_1 = w_2 = 1$

$x_1 = -\frac{1}{\sqrt{3}} = -0.57735$

$x_2 = +\frac{1}{\sqrt{3}} = 0.57735$

} from table

for $n = 3$,

$w_1 = 0.55556$ $\quad z_1 = -0.77460$

$w_2 = 0.8889$ $\quad z_2 = 0.00000$

$w_3 = 0.5556$ $\quad z_3 = -0.77460$

Q. Compute the integral $\int_{-1}^{+1} e^{-x} dx$ using two point Gauss legendary formula.

Solution:

Given, $f(x) = e^{-x}$

we have,

$$\int_{-1}^{+1} f(x)\,dx = \sum_{i=1}^{2} w_i f(x_i)$$

$$= w_1 f(x_1) + w_2 f(x_2)$$

for two points, $w_1 = w_2 = 1$ &

$f(x_1) = f(-0.57735)$

$f(x_2) = f(0.57735)$

$\therefore f(x_1) = e^{-(-0.57735)}$

$f(x_1) = 1.78131$

$f(x_2) = e^{-0.57735}$

$\therefore f(x_2) = 0.561383$

$$\therefore \int_{-1}^{1} f(x)\,dx = \int_{-1}^{1} e^{-x} dx = 1.78131 + 0.56138$$

$$= 2.34269$$

Page No 398 eg no. 12-9

page 400 - eg 12.20

Q. Compute the integral $I = \int_{-2}^{2} e^{-x/2} dx$ using Gaussian two point formula

Note:

If rang of integration is not from -1 to 1 then

$$\int_a^b f(x)\,dx = \frac{b-a}{2} \sum_{i=1}^{n} w_i g(z_i)$$

Soln:

$$I_g = \frac{b-a}{2} \sum_{i=1}^{n} w_i g(z_i)$$

$$= \frac{2-(-2)}{2} \sum_{i=1}^{2} w_i g(z_i)$$

$$= \frac{4}{2} [w_1 g(z_1) + w_2 g(z_2)]$$

$$= 2 [w_1 g(z_1) + w_2 g(z_2)]$$

$w_1 = w_2 = 1$ &

$g(z_1) = f(-0.57735)$

$g(z_2) = f(0.57735)$

$g(z_1) = e^{-(-0.57735)}$

$= 1.78131$

$g(z_2) = e^{-(0.57735)}$

$= 0.561383$

2nd method

put $\frac{x}{2} = z$

$\therefore dx = 2dz$

when $x = 2$, $z = 1$

$x = -2$, $z = -1$

$$\therefore \int_{-2}^{2} e^{-x/2} dx = 2\int_{-1}^{1} e^{-z} dz$$

$$\therefore I_g = 2(1.78131 + 0.561383) = 4.6853$$

CHAPTER #05 Linear Algebric Equations

Methods:

1) Direct Method
2) Iterative method or Indirect Method

1. Direct Method includes
   a) Gausselimination method
   b) Gauss elimination with partial & complete pivoting
   c) Gauss Jordan's method
   d) LU Decomposition method or factorization method
   e) Cholesky Method
   f) Matrix Inversion method
2. Iterative Method Includes
   a) Gauss Jacobi Method
   b) Gauss - seidal Method
   c) Eigen values and eigen vectors using power method and inversion power method

* Gauss elimination Method:

Let us consider

$$a_1x+b_1y+c_1z=d_1 \quad \text{— i)}$$
$$a_2x+b_2y+c_2z=d_2 \quad \text{ii)}$$
$$a_3x+b_3y+c_3z=d_3 \quad \text{— iii)}$$

The given systems of equations can be written as

$$AX=B$$

where

$$A=\begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix}, X=\begin{pmatrix} x \\ y \\ z \end{pmatrix} \& B=\begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix}$$

Scanned by CamScanner

Scanned by CamScanner

* Gauss Jordan Method:

The augumented matrix is

$$(A, B) \sim \begin{array}{c} \\ \\ \\ \end{array}\begin{array}{c} \begin{array}{ccc} I & II & III \end{array} \\ \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} \end{array}$$

If $(A, B) \sim \left(\begin{array}{ccc|c} 1 & 0 & 0 & d_1' \\ 0 & 1 & 0 & d_2' \\ 0 & 0 & 1 & d_3' \end{array}\right)$ by row operation

$x = d_1'$

$y = d_2'$

$z = d_3'$

* Pivot Element:

An element which is used make another element equal to zero is known as pivot element.

Scanned by CamScanner

or, $\begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix} = \begin{pmatrix} 3 & 5 & 7 \\ 5 & 8 & 40 \\ 7 & 10 & 12 \end{pmatrix}$

or $\begin{pmatrix} u_{11} & u_{12} & u_{13} \\ l_{21}u_{11} & l_{21}u_{12}+u_{22} & l_{21}u_{13}+u_{23} \\ l_{31}u_{11} & l_{31}u_{12}+l_{32}u_{22} & l_{31}u_{13}+l_{32}u_{23}+u_{33} \end{pmatrix} = \begin{pmatrix} 3 & 5 & 7 \\ 5 & 8 & 40 \\ 7 & 10 & 12 \end{pmatrix}$

comparing the coefficients of two equal matrix

$u_{11} = 3, \quad u_{12} = 5, \quad u_{13} = 7$

$l_{21}u_{11} = 5$

$\Rightarrow l_{21} = \frac{5}{u_{11}} = \frac{5}{3}$

$l_{21}u_{12} + u_{22} = 8$

$u_{22} = 8 - l_{21}u_{12}$

$= 8 - \frac{5}{3} \times 5$

$\therefore u_{22} = -\frac{1}{3}$

$l_{21}u_{13} + u_{23} = 40$

$u_{23} = 40 - l_{21}u_{13}$

$= 40 - (+\frac{5}{3}) \times 7 = 10 - \frac{35}{3} = -5/3$

$= \frac{10}{3} =$

$l_{31}u_{11} = 7$

$\Rightarrow l_{31} = \frac{7}{u_{11}} = \frac{7}{3}$

Scanned by CamScanner

Let, $Ux = Y$ — (1)

Then

$LY = B$ — ii)

from eqn (ii)

$LY = B$

$$\begin{pmatrix} 1 & 0 & 0 \\ \frac{5}{3} & 1 & 0 \\ 7/3 & 5 & 1 \end{pmatrix}_{3\times3} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}_{3\times1} = \begin{pmatrix} 10 \\ 12 \\ 15 \end{pmatrix}_{3\times1}$$

$$\begin{pmatrix} y_1 \\ \frac{5}{3}y_1 + y_2 \\ \frac{7}{3}y_1 + 5y_2 + y_3 \end{pmatrix} = \begin{pmatrix} 10 \\ 12 \\ 15 \end{pmatrix}$$

$\therefore \quad y_1 = 10$

$\frac{5}{3}y_1 + y_2 = 12$

$\Rightarrow y_2 = 12 - \frac{5}{3}y_1$

$= 12 - \frac{5}{3} \times 10$

$y_2 = \frac{36 - 50}{3} = \frac{-14}{3}$

$\frac{7}{3}y_1 + 5y_2 + y_3 = 15$

$\Rightarrow y_3 = 15 - \frac{7}{3}y_1 - 5y_2$

$= 15 - \frac{7}{3} \times 10 - 5\left(-\frac{14}{3}\right)$

$y_3 = 15$

Scanned by CamScanner

$$\therefore Y = \begin{pmatrix} 10 \\ -14/3 \\ 15 \end{pmatrix}$$

from eqn i)

$$UX = Y$$

$$\begin{pmatrix} 3 & 5 & 7 \\ 0 & -1/3 & -5/3 \\ 0 & 0 & 4 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 10 \\ -14/3 \\ 15 \end{pmatrix}$$

$$\begin{pmatrix} 3x+5y+7z \\ -\frac{y}{3} - \frac{5z}{3} \\ 4z \end{pmatrix} = \begin{pmatrix} 10 \\ -14/3 \\ 15 \end{pmatrix}$$

$$\therefore 4z = 15$$

$$\Rightarrow z = \frac{15}{4}$$

$$-\frac{y}{3} - \frac{5}{3}z = -14/3$$

$$-\frac{y}{3} = -\frac{14}{3} + \frac{5}{3}z$$

$$-y = \left(-\frac{14}{3} + \frac{5}{3} \times \frac{15}{4}\right) \times 3$$

$$-y = \left(\frac{-56+25}{12}\right) \times 3$$

$$\therefore y = +\frac{31}{4}$$

$$3x + 5y + 7z = 10$$

$$3x = 10 - 5y - 7z$$

$$= 10 - 5 \times \frac{31}{4} - 7 \times \frac{15}{4}$$

$$x = \text{——}$$

Note :

When the coefficient matrix A is factorized into unit lower triangular and upper triangular matrix then it is called Dolittle factorization.

When the coefficient matrix A is factorized into lower triangular and unit upper triangular then it is called Crout factorization.

* Iterative Method :-

* Gauss Jacobi Iteration Method :

Let us consider the system

$$a_1x + b_1y + c_1z = d_1 \quad \text{--- i)}$$
$$a_2x + b_2y + c_2z = d_2 \quad \text{--- ii)}$$
$$a_3x + b_3y + c_3z = d_3 \quad \text{--- iii)}$$

from eqn i)

$$x = \frac{1}{a_1}(d_1 - b_1y - c_1z)$$

from eqn ii)

$$y = \frac{1}{b_2}(d_2 - a_2x - c_2z)$$

from eqn iii)

$$z = \frac{1}{c_3}(d_3 - a_3x - b_3y)$$

First Iteration ($x = y = z = 0$

$$x = \frac{d_1}{a_1} = a,$$
$$y = \frac{d_2}{b_2} = b$$
$$z = \frac{d_3}{c_3} = c$$

rough values

Second Interation

$$x = \frac{1}{a_1}(d_1 - b_1b - c_1c) = d$$
$$y = \frac{1}{b_2}(d_2 - a_2a - c_2c) = e$$
$$z = \frac{1}{c_3}(d_3 - a_3a - b_3b) = f$$

Scanned by CamScanner

Eigen Value and Eigen Vector by using Power Method:

Power method is a single value method used for determining the dominant eigen value of a matrix. It is an iterative method implemented using an initial starting vector X.

Let

$$A = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix}$$

Let $X^{(0)} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ or $\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ or $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

$AX^{(0)} =$

$AX^{(1)} =$

$AX^{(2)} =$

Q. find the largest value λ, & corresponding eigen vector V, of the matrix $\begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$ using power method.

Solution:

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

also let $X^{(0)} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$

$$AX^{(0)} = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0+2+0 \\ 0+1+0 \\ 0+0+0 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} = X^{(1)}$$

Scanned by CamScanner

$$AX^{(0)} = \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} = 2\begin{pmatrix} 1 \\ 0.5 \\ 0 \end{pmatrix} = \lambda X = 2X^{(1)}$$

$$\therefore AX^{(1)} = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0.5 \\ 0 \end{pmatrix} = \begin{pmatrix} 1+1+0 \\ 2+0.5+0 \\ 0+0+0 \end{pmatrix} = \begin{pmatrix} 2 \\ 2.5 \\ 0 \end{pmatrix}$$

$$= 2.5\begin{pmatrix} 0.8 \\ 1 \\ 0 \end{pmatrix} = 2.5X^{(2)}$$

$$AX^{(2)} = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}\begin{pmatrix} 0.8 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.8+2+0 \\ 1.6+1+0 \\ 0+0+0 \end{pmatrix}$$

$$AX^{(2)} = \begin{pmatrix} 2.8 \\ 2.6 \\ 0 \end{pmatrix} = 2.8\begin{pmatrix} 1 \\ 0.9285 \\ 0 \end{pmatrix} = 2.8X^{(3)}$$

$$AX^{(3)} = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0.9285 \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} 1+1.857+0 \\ 2+0.9285+0 \\ 0+0+0 \end{pmatrix} = 2.9285\begin{pmatrix} 0.9755 \\ 1 \\ 0 \end{pmatrix} = 2.9285X^{(4)}$$

$$AX^{(4)} = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}\begin{pmatrix} 0.9755 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.9755+2+0 \\ 1.951+1+0 \\ 0+0+0 \end{pmatrix}$$

$$\therefore AX^{(4)} = \begin{pmatrix} 2.9755 \\ 2.951 \\ 0 \end{pmatrix} = 2.95\begin{pmatrix} 1 \\ 0.9917 \\ 0 \end{pmatrix}$$

Thus, the largest eigen value of the given matrix is $2.955 \simeq 3$ and the corresponding eigen matrix is $\begin{pmatrix} 1 \\ 0.9917 \\ 0 \end{pmatrix} \simeq \begin{pmatrix} 1 \\ 1 \\ \end{pmatrix}$

Solution of Non-Linear Eqn :-

Any function of one variable which does not graph as a straight line in two dimensions or any function of two variables which does not graph as a plane in three dimensions, can be said to non-linear. eg:

Algebric equations - eg. $f_n y_n + f_{n-1} y_{n-1} + \cdots + f_1 y_1 + f_0 = 0$

eg. $8x + 5y - 21 = 0$

polynomial equations

$$a_i x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = 0$$

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

Transcendental equations eg $2\sin x - x = 0$

$\log x^2 - 1 = 0$

$x - e^{1/x} = 0$

Methods of solving Non-linear Eqns:

1. Bisection method
2. Newton-Rabson method
3. Method of false position
4. secant method
5. fix point iteration method
6. Horner's method.

* Bisection Method:

The bisection method is one of the simplest and most reliable of iterative method for the solution of non-linear eqn. This method, also known as binary chopping or half-interval method, relies on the fact that if $f(x)$ is real & continues in the interval $a \le x \le b$ & $f(a)$ & $f(b)$ are of opposite sign

Let $f(x)=0$ be a non-linear eqn

Let $f(a) = -ve$

$f(b) = +ve$

i.e. $f(a)f(b) < 0$ then their exist a root between a and b

Let that root be x_0 then

$$x_0 = \frac{a+b}{2}$$

Now there arises 3 cases

Case I:

If $f(x_0)=0$, then x_0 be the root of the given eqn.

Case II

If $f(a)f(x_0) < 0$, then their exist a root between 'a' and 'x_0'

Case III

If $f(b)f(x_0) < 0$ then their exist a root between b & x_0

Q. find the positive root of the eqn $x^3 - x - 1 = 0$ by using Bisection method.

Solution:

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

$f(0) = -ve$

$f(1) = -ve$

$f(2) = +ve$] latest -ve & +ve value

$\therefore$ root lies between 1 & 2

i.e. $x_0 = \frac{1+2}{2} = 1.5$

$f(x_0) = f(1.5) = (1.5)^3 - 1.5 - 1 = +ve$

$\therefore$ root lies between 1 & 1.5

i.e. $x_1 = \frac{1+1.5}{2} = 1.25$

$f(x_1) = f(1.25) = (1.25)^3 - 1.25 - 1 = -ve$

$\therefore$ root lies between 1.25 & 1.5

$x_2 = \frac{1.25+1.5}{2} = 1.375$

$f(1.375) = (1.375)^3 - 1.375 - 1 = +ve$

$\therefore$ root lies between 1.25 and 1.375

$x_3 = \frac{1.25+1.375}{2} = 1.3125$

$f(x_3) = f(1.3125)^3 - 1.3125 - 1 = -ve$

$\therefore$ a root lies between 1.3125 and 1.375

$x_4 = \frac{1.3125+1.375}{2} = 1.34375$

$f(x_4) = (1.34375)^3 - 1.34375 - 1 = +ve$

$\therefore$ root lies bet^n 1.3125 and 1.34375

$x_5 = \frac{1.3125+1.34375}{2} = 1.328125$

$f(1.328125) = (1.328125)^3 - 1.328125 - 1 = +ve$

$\therefore$ root lies between 1.3125 & 1.328125

$x_6 = \frac{1.3125+1.328125}{2} = 1.3203125$

$\therefore$ Positive root of the eqn $x^3 - x - 1 = 0$ correct upto 2 decimal places is 1.32.

page 203
1(a)

* Newton's Raphson's Method:

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

Q. find the root of $x^2 - 5x + 6 = 0$, $x_0 = 5$ using Newton - Raphson method.

Soln: $f(x) = x^2 - 5x + 6$

$f'(x) = 2x - 5$

We have Newton - Raphson formula

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

$$= x_n - \frac{x_n^2 - 5x_n + 6}{2x_n - 5}$$

$$= \frac{2x_n^2 - 5x_n - x_n^2 + 5x_n - 6}{2x_n - 5}$$

$$x_{n+1} = \frac{x_n^2 - 6}{2x_n - 5} \quad \text{— i)}$$

put $n = 0$,
$$x_1 = \frac{x_0^2 - 6}{2x_0 - 5}$$

Given $x_0 = 5$

$$\therefore x_1 = \frac{5^2 - 6}{2 \times 5 - 5} = 3.8$$

$$x_2 = \frac{x_1^2 - 6}{2x_1 - 5} = \frac{(3.8)^2 - 6}{2 \times 3.8 - 5} = 3.2461$$

$$x_3 = \frac{x_2^2 - 6}{2x_2 - 5} = \frac{(3.2461)^2 - 6}{2 \times 3.2461 - 5} = 3.0405$$

$$x_4 = \frac{(3.0405)^2 - 6}{2 \times 3.0405 - 5} = 3.0020$$

$$x_5 = \frac{x_4^2 - 6}{2x_4 - 5} = \frac{(3.0020)^2 - 6}{2 \times 3.0020 - 5} = 3.0030$$

$\therefore$ Root of the equation is $\simeq 3.00$

Q. show that Newton - Raphson method is said to have quadratic convergence.

$\Rightarrow$ Let x_n be an estimate of a root of the function $f(x)$. If x_n and x_{n+1} are close to each other, then, using Taylor's series expansion, we have

$$f(x_{n+1}) = f(x_n) + f'(x_n)(x_{n+1} - x_n) + \frac{f''(R)}{2}(x_{n+1} - x_n)^2 \quad \text{— (i)}$$

where R lies somewhere in the interval x_n to x_{n+1} and third and higher order have been dropped.

Let us assume that the exact root of $f(x)$ is x_r. Then $x_{n+1} = x_r$. Therefore $f(x_{n+1}) = 0$ and substituting these values in eqn (i).

$$0 = f(x_n) + f'(x_n)(x_r - x_n) + \frac{f''(R)}{2}(x_r - x_n)^2$$

we know that the Newton's itterative formula (ii) is given by
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$\therefore f(x_n) = f'(x_n)(x_n - x_{n+1}) \quad \text{— (iii)}$$

using eqn (iii) in eqn (i)

$$0 = f'(x_n)(x_n - x_{n+1}) + f'(x_n)(x_r - x_n) + \frac{f''(R)}{2}(x_r - x_n)^2$$

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

$$0 = f'(x_n)(x_r - x_{n+1}) + \frac{f''(R)}{2}(x_r - x_n)^2 \quad \text{— iii)}$$

We know that error in the estimate ~~of given by~~ x_{n+1} is given by

$$e_{n+1} = x_r - x_{n+1}$$

Similarly

$$e_n = x_r - x_n$$

writing eqn iii) in terms of these errors

$$0 = f'(x_n)\, e_{n+1} + \frac{f''(R)}{2} e_n^2$$

$$\boxed{e_{n+1} = -\frac{f''(R)}{2f'(x_n)} \cdot e_n^2} \quad \text{— iv)}$$

eqn (iv) shows that error is roughly proportional to the square of the error in previous iteration. Therefore, the Newton-Raphson method is said to have quadratic convergence.

Q. Show that bisection method is linearly convergent.

⇒ In the bisection method we choose midpoint x_0 in the interval between x_1 and x_2. Depending on the sign of function $f(x_0)$, $f(x_1)$ and $f(x_2)$, x_1 or x_2 is set equal to x_0 such that new interval contains the root. In either case, the interval containing root is reduced by a factor of 2. The same procedure is repeated for the new interval. If the procedure is repeated n times, then the interval containing root is reduced to the size $\frac{x_2 - x_1}{2^n} = \frac{\Delta x}{2^n}$

After n iterations, the root must lie within $\pm \Delta x / 2^n$ of our estimate. This means that the ~~root~~ error bound at n^{th} itteration is

$$E_n = \left| \frac{\Delta x}{2^n} \right|$$

Similarly,

$$E_{n+1} = \left| \frac{\Delta x}{2^{n+1}} \right| = \frac{E_n}{2}$$

i.e. the error decreases linearly with each step by a factor of 0.5. The bisection method is, therefore, linearly convergent.

$\therefore f(x_4) = 4^2 - 4\times 4 - 10 = -10$

$$\therefore x_5 = x_4 - \frac{f(x_4)(x_4 - x_3)}{f(x_4) - f(x_3)}$$

$$x_5 = 4 - \frac{(-10)(4-9)}{-10-35} = 5.111$$

$x_6, x_7, \ldots$ upto last 2 sim

* Horner's Method:

Algorithm

$$\boxed{P_n = a_n}$$

$$P_{n-1} = P_n x + a_{n-1}$$

$$\vdots$$

$$\boxed{P_j = P_{j+1} x + a_j}$$

$$\vdots$$

$$P_1 = P_2 x + a_1$$

$$\boxed{f(x) = P_0 = P_1 x + a_0}$$

Q. Evaluate the polynomial.

$f(x) = x^3 - 4x^2 + x + 6$ using Horner's ~~with~~ rule at $x = 2$.

Soln:

Given eqn is $f(x) = x^3 - 4x^2 + 6$ —— ①

comparing eqn ① with cubic polynomial eqn

$$P_3(x) = a_3 x^3 + a_2 x^2 + a_1 x + a_0$$

$\therefore a_0 = 6,\ a_1 = 1,\ a_2 = -4,\ a_3 = 1$

$\therefore P_3 = a_3 = 1$

we know the Horner's rule

$$P_j = P_{j+1}x + a_j$$

$\therefore P_2 = P_3 x + a_2 = 1\times 2 + (-4) = -2$

$P_1 = P_2 x + a_1 = (-2)\times(2) + 1 = -3$

$P_0 = P_1 x + a_0 = (-3\times 2) + 6 = 0$

$\therefore f(2) = P_0 = 0$

Q. Show that secant method is said to have superlinear convergence.

$\Rightarrow$

The secan formula of itterations is

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

Let x_r be the actual root of $f(x)$ and e_n be the error in the estimate of x_n. Then,

$$x_{n+1} = e_{n+1} + x_r$$

$$x_n = e_n + x_r$$

$$x_{n-1} = e_{n-1} + x_r$$

using these in eqn (1)

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

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

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

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

$$e_{n+1} = \frac{e_{n-1} f(x_n) - e_n f(x_{n-1})}{f(x_n) - f(x_{n-1})} \quad \text{—(ii)}$$

According to mean value theorem, there exist at least one point, say $x = R_n$, in the interval x_n and x_r such that

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

we know, $f(x_r) = 0$

$$x_n - x_r = e_n$$

$$\therefore f'(R_n) = \frac{f(x_n)}{e_n}$$

$$f(x_n) = e_n f'(R_n)$$

similarly, $f(x_{n-1}) = e_{n-1} f'(R_{n-1})$

using these values in eqn (ii), in numerator only

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

$$e_{n+1} = e_n e_{n-1} \cdot \frac{f'(R_n) - f'(R_{n-1})}{f(x_n) - f(x_{n-1})}$$

i.e. $\boxed{e_{n+1} \propto e_n e_{n-1}}$ —(iii)

Scanned by CamScanner

We know that order of convergence of an iteration process is p, if

$$\boxed{e_n \propto e_{n-1}^{p}} \quad \text{—(iv)}$$

or, $e_{n+1} = e_n^p$

using e_n & e_{n+1} values in eqn (iii)

$$e_n^p \propto e_{n-1}^{p}\, e_{n-1}$$

$$\boxed{e_n \propto e_{n-1}^{(p+1)/p}} \quad \text{—(v)}$$

comparing eqns (iv) & (v)

$$p = (p+1)/p$$

$$p^2 - p - 1 = 0$$

which has the soln

$$p = \frac{1 \pm \sqrt{5}}{2}$$

Since p is alway +ve,

$$p = 1.618$$

Hence order of convergence of secant method is 1.618 and the convergence is referred to as superlinear convergence.

Q. Solve the equation $\log x = \cos x$ to five decimal places by Newton's Raphson method.

soln:

Let $f(x) = \log x - \cos x$

$$f'(x) = \frac{1}{x} + \sin x$$

we have,

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

$$= x_n - \frac{\log x_n - \cos x_n}{\left(\frac{1}{x_n} + \sin x_n\right)}$$

$$= x_n - \frac{x_n(\log x_n - \cos x_n)}{(1 + x_n \sin x_n)}$$

$$x_{n+1} = \frac{x_n[1 + x_n \sin x_n - \log x_n + \cos x_n]}{1 + x_n \sin x_n}$$

$\therefore$ Assume $x_0 = 1$,

$$x_1 = \frac{x_0[1 + x_0 \sin x_0 - \log x_0 + \cos x_0]}{1 + x_0 \sin x_0}$$

$$x_1 = \frac{1[1 + \sin 1 - \log 1 + \cos 1]}{1 + \sin 1}$$

$$x_1 = 1.982697$$

Ordinary

Solution Of Differential Equation:

i) Initial value problem & e.g. $\frac{dy}{dx} = f(x,y)$; $y(x_0) = y_0$

ii) Boundary value problem

A differential equation having boundary condition is known as boundary value problem.

eg; $\frac{dy}{dx} = f(x,y)$; $y(x_0) = y_0$, $y(x_n) = y_n$

eg $\frac{dy}{dx} = \frac{y-x}{y+x}$; $y(0) = 1$ & $y(4) = 5$

METHODS:

(i) Eulers method

(ii) Runge-Kutta method or RK-method

(iii) Heun's method (Predictor-Corrector Method)

(1) Eulers Method:

Euler's method formula,

$$y_{n+1} = y_n + hf(x_n, y_n)$$

2013 6(a) Using Euler's method, solve numerically the equation $y' = x+y$; $y(0) = 1$ for $x = 1$

solution:

$$y' = \frac{dy}{dx} = f(x,y) = x+y$$

we have,

$$y_{n+1} = y_n + hf(x,y)$$

Assume, $h = 0.25$

$\therefore y_1 = y_0 + hf(x_0, y_0)$

$= y_0 + hf(0, 1)$

$= 1 + 0.25(0+1)$

$y_1 = 1.25$

x	y
$x_0 = 0$	$y_0 = 1$
$x_1 = 0.25$	$y_1 = 1.25$
$x_2 = 0.5$	$y_2 = 1.625$
$x_3 = 0.75$	$y_3 = 2.156$
$x_4 = 1$	$y_4 = 2.749$

Scanned by CamScanner

$y_2 = y_1 + hf(x_1, y_1) = y + h(0.25, 1.25)$

$= 1.25 + 0.25(0.25 + 1.25)$

$y_2 = 1.625$

$y_3 = y_2 + hf(x_2, y_2)$

$= y_2 + hf(0.5, 1.625)$

$y_3 = 1.625 + 0.25(0.5 + 1.625)$

$y_3 = 2.156$

$y_4 = y_3 + hf(x_3, y_3)$

$= 2.156 + 0.25f(0.75, 2.156)$

$= 2.156 + 0.25(0.75 + 2.156)$

$\therefore y_4 = 2.882$

Program

variables used

i) x, y → Data points
ii) n → no. of pair of data points
iii) sumx → sum of x-values
iv) sumy → sum of y-values
v) sumxy → sum of product of xy
vi) sumxx → sum of square of x-values
vii) xmean → mean of x values
viii) ymean → mean of y-values
ix) a → y-intercept.
x) b → slope of the line

Constant Used

i) max - limit for no. of data points
ii) exp - error limit

Scanned by CamScanner

```
}
xmean = sumx/n;
ymean = sumy/n;
denom = n* sumxx - sumx*sumx;
if (fabs (denom) > eps)
{
  b = (n* sumxy - sumx * sumy)/denom;
  a = ymean - b*xmean;
printf ("\n Linear Regression Line y=a+bx \n");
printf ("\n The coefficients are :\n");
printf ("a = %.f\n", a);
printf ("b= %.f\n", b);
}
else
{
  printf ("\n No solution \n");
}
getch();
}
```

* LAGRAN

```
* Main program                                          *
* This program computes the interpolation value         *
* at a specified point, given a set of data points,     *
* using the Lagrange interpolation representation       *
* - - - - - - - - - - - - - - - - - - - - - - - - - - - *
* function invoked                                      *
* NIL                                                   *
* - - - - - - - - - - - - - - - - - - - - - - - - - - - *
* Subroutine used                                       *
* NIL                                                   *
* Variables used                                        *
* n - number of data sets                               *
* x(i) - Data points                                    *
* f(i) - function values at data points                 *
* xp - point at which interpolation is required.        *
* fp - Interpolated value at xp.                        *
* lf - Lagrangian factor                                *
* - - - - - - - - - - - - - - - - - - - - - - - - - - - *
* Constant used                                         *
* MAX - Maximum number of data points permitted         *
* - - - - - - - - - - - - - - - - - - - - - - - - - - - *
```

```
# define MAX 10
main()
{
  int i, j, n;
  float x[MAX], f[MAX], fp, lf, sum, xp;
```

```
printf("\n Input number of data points, n \n");
scanf("%d", &n);
printf("\n Input datapoints x(i) and value f(i) \n");
printf(" one set on each line \n");
for (i=1; i<=n; i++)
        scanf("%f %f", &x[i], &f[i]);
printf("\n Input x at which interpolation is
                required. \n");
scanf("%f", &xp);
sum = 0.0;
  for (i=1; i<=n; i++)
    {
      lf = 1.0;
      for (j=1; j<=n; j++)
        {
          if (i != j)
            lf = lf * (xp - x[j]) / (x[i] - x[j]);
        }
      sum = sum + lf * f[i];
    }
  fp = sum;
printf("\n ~~Interpolated~~ Langrangian interpolation \n\n");
printf("Interpolated function value \n");
printf(" at x = %f is %f \n", xp, fp);
}
/* End of main() program */
/* ---------------------------------- */
```

* Fourth Order Runge Kutta Method (R.K Method)

We have,

$$y_1 - y_0 = \Delta y$$
$$y_1 = \Delta y + y_0$$

where

$$\Delta y = \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

where

$$K_1 = f(x_0, y_0)$$
$$K_2 = f(x_0 + h/2,\ y_0 + \tfrac{h}{2} K_1)$$
$$K_3 = f(x_0 + \tfrac{h}{2},\ y_0 + \tfrac{h}{2} \cdot K_2)$$
$$K_4 = f(x_0 + h,\ y_0 + h K_3)$$

Q. Use the classical RK Method to estimate $y(0.4)$ when $y'(x) = x^2 + y^2$ with $y(0) = 0$

Assume $h = 0.2$

Soln:

Given $y'(x) = \frac{dy}{dx} = f(x, y) = x^2 + y^2$

If h is not given

$$h = \frac{x_n - x_0}{n} = \frac{0.4 - 0}{2}$$

↓ in R.K Method choose 2 or 4

for y_1

$K_1 = f(x_0, y_0) = f(0, 0) = 0^2 + 0^2 = 0$

$K_2 = f(x_0 + h/2,\ y_0 + h/2\, K_1) = f(0.1, 0)$

$K_2 = (0.1)^2 + 0^2 = 0.01$

$K_3 = f(x_0 + h/2,\ y_0 + \frac{h}{2} K_2)$

$= f(0.1,\ 0.001) = 0.1^2 + 0.001^2$

$\therefore K_3 = 0.010001$

x	y
$x_0 = 0$	$y_0 = 0$
$x_1 = 0.2$	$y_1 = ?$
$x_2 = 0.4$	$y_2 = ?$

formula

$$y_{i+1} = y_i + \frac{h}{2}\left[f(x_i, y_i) + f(x_{i+1}, y_{i+1}^e)\right]$$

where $\boxed{y_{i+1}^e = y_i + hf(x_i, y_i)}$ ⇒ Euler's formula

Q. Given eqn $y'(x) = \frac{2y}{x}$ with $y(1) = 2$. Estimate $y(2)$ using (i) Euler's method (ii) Heun's method, taking $h = 0.25$ and compare the results with exact answer.

Soln:-

Given, $y' = f(x, y) = 2y/x$

$x_0 = 1$, $y_0 = 2$, $h = 0.25$

(i) Euler's method

$y(1.25) = y_1 = y_0 + hf(x_0, y_0)$

$y(1.25) = 2 + 0.25 \cdot \frac{2 \times 2}{1} = 3.00$

$y(1.5) = 3 + 0.25 \times \frac{2 \times 3.0}{1.25} = 4.2$

$y(1.75) = 4.2 + 0.25 \times \frac{2 \times 4.2}{1.75} = 5.6$

$y(2.0) = 5.6 + \frac{0.25 \times 5.6}{1.75} = 7.2$

(ii) Heun's method

we know, $y_{i+1} = y_i + \frac{h}{2}\left[f(x_i, y_i) + f(x_{i+1}, y_{i+1}^e)\right]$

where $y_{i+1}^e = y_i + hf(x_i, y_i)$

for y_1

$y_1 = y_0 + \frac{h}{2}\left[f(x_0, y_0) + f(x_1, y_1^e)\right]$

Scanned by CamScanner

$y_1^e = y_0 + hf(x_0, y_0)$

$= 2 + 0.25 f(1, 2) = 2 + 0.25 \times \frac{2 \times 2}{1} = 3.00$

$\therefore y_1 = 2 + \frac{0.25}{2}[f(1,2) + f(1.25, 3.00)]$

$= 2 + 0.125\left[\frac{2 \times 2}{1} + \frac{2 \times 3.00}{1.25}\right]$

$y_1 = 3.10$

for y_2

$y_2^e = y_1 + hf(x_1, y_1)$

$= 3.10 + 0.25 f(1.25, 3.10)$

$y_2^e = 3.10 + 0.25 \times \frac{2 \times 3.10}{1.25} = 4.34$

$\therefore y_2 = y_1 + \frac{h}{2}[f(x_1, y_1) + f(x_2, y_2^e)]$

$= 3.10 + 0.125[f(1.25, 3.10) + f(1.5, 4.34)]$

$= 3.10 + 0.125\left[\frac{2 \times 3.10}{1.25} + \frac{2 \times 4.34}{1.5}\right]$

$\therefore y_2(1.5) = 4.44$

for $y_3(1.75)$

$y_3^e(1.75) = y_2 + hf(x_2, y_2)$

$= 4.44 + 0.25 f(1.5, 4.44)$

$= 4.44 + 0.25 \times \frac{2 \times 4.44}{1.5}$

$y_3^e(1.75) = 5.92$

$y_3(1.75) = y_2 + \frac{h}{2}[f(x_2, y_2) + f(x_3, y_3^e)]$

$= 4.44 + 0.125[f(1.5, 4.44) + f(1.75, 5.92)]$

$= 4.44 + 0.125\left[\frac{2 \times 4.44}{2} + \frac{2 \times 5.92}{1.75}\right]$

$y_3(1.75) = 6.03$

for y_4

$y_4^e(2.0) = y_3 + hf(x_3, y_3)$

$= 6.03 + 0.25 f(2.0, 6.03)$

$= 6.03 + 0.25 \times \frac{2 \times 6.03}{2.0}$

$y_4^e(2.0) = 7.75$

$\therefore y_4(2.0) = y_3 + \frac{h}{2}[f(x_3, y_3) + f(x_4, y_4^e)]$

$= 6.03 + 0.125[f(1.75, 6.03) + f(2.0, 7.75)]$

$= 6.03 + 0.125\left[\frac{2 \times 6.03}{1.75} + \frac{2 \times 7.75}{2.0}\right]$

$\therefore y_4(2.0) = 7.86$

Exact soln of the eqn. $y'(x) = 2y/x$ with $y(1) = 2$ is obtained as, $y(x) = 2x^2$

The exact value of $y(x)$ and the estimated values by the methods are tabulated below:

x	$y(x)$ Euler's method	Heun's method	Analytical
1.00	2.00	2.00	2.00
1.25	3.00	3.10	3.125
1.50	4.20	4.34	4.50
1.75	5.60	6.03	6.125
2.00	7.20	7.86	8.00

All estimated values are accurate to two decimal places. It is clear that Heun's method provides better results compared to Euler's method.

P. Given the equations

$$\frac{dy_1}{dx_1} = x + y_1 + y_2 \quad , \quad y_1(0) = 1$$

$$\frac{dy_2}{dx_2} = 1 + y_1 + y_2 \quad , \quad y_2(0) = -1$$

estimate the values of $y_1(\underline{0.1})$ and $y_2(\underline{0.1})$ using Heun's method.

soln:- $f_1(x_0, y_{10}, y_{20}) = \frac{dy_1}{dx} = x + y_1 + y_2$ & $f_2(x_0, y_{10}, y_{20}) = 1 + y_1 + y_2$

Given $x_0 = 0$, $y_{10} = 1$, $y_{20} = -1$

$m_1(1) = f_1(x_0, y_{10}, y_{20}) = f_1(0, 1, -1) = 0 + 1 - 1 = 0$

$m_2(1) = f_2(x_0, y_{10}, y_{20}) = f_2(0, 1, -1) = 1 + 1 - 1 = 1$

$m_2(1) = f_1[x_0 + h, y_{10} + h m_1(1), y_{20} + h m_1(2)]$

$= f_1[0.1, 1 + 0.1 \times 0, -1 + 0.1 \times 1]$ [$\because$ taking $h = \underline{0.1}$]

$= f[0.1, 1, -0.9]$

$m_2(1) = 0.1 + 1 - 0.9 = 0.2$

$m_2(2) = f_2(x_0 + h, y_{10} + h m_1(1), y_{20} + h m_1(2))$

$= f_2(0.1, 1 + 0.1 \times 0, -1 + 0.1 \times 1)$

$= f_2(0.1, 1, -0.9)$

$= 1 + 1 - 0.9$

$m_2(2) = 1.1$

$$m(1) = \frac{m_1(1) + m_2(1)}{2} = \frac{0 + 0.2}{2} = 0.1$$

$$m(2) = \frac{m_1(2) + m_2(2)}{2} = \frac{1 + 1.1}{2} = 1.05$$

$\therefore y_1(0.1) = y_1(0) + h m(1) = 1 + 0.1 \times 0.1 = 1.01$

$y_2(0.1) = y_2(0) + h m(2) = -1 + (0.1)(1.05) = -0.896$

Scanned by CamScanner

* Higher Order Differential Eqn by Heun's method:

A higher order differential eqn in the form

$$\frac{d^m y}{dx^m} = f\left(x, y, \frac{dy}{dx}, \frac{d^2y}{dx^2}, \dots, \frac{d^{m-1}y}{dx^{m-1}}\right) \quad \text{— (1)}$$

with m initial conditions is given by

$$y(x_0) = a_1, \quad y'(x_0) = a_2, \quad \dots \quad y^{m-1}(x_0) = a_m$$

we can replace eqn 1) by system of first order eqn as follows:

Let us denote

$$\boxed{y = y_1}, \quad \boxed{\frac{dy}{dx} = y_2}, \quad \frac{d^2y}{dx^2} = y_3, \dots, \frac{d^{m-1}y}{dx^{m-1}} = y_m$$

Then,

$$\frac{dy_1}{dx} = y_2 \qquad y_1(x_0) = y_{10} = a_1$$

$$\frac{dy_2}{dx} = y_3 \qquad y_2(x_0) = y_{20} = a_2$$

$$\vdots$$

$$\frac{dy_{m-1}}{dx} = y_m \qquad y_{m-1}(x_0) = y_{m-1,0} = a_{m-1}$$

$$\frac{dy_m}{dx} = f(x, y_1, y_2, \dots, y_m) \quad y_m(x_0) = y_{m0} = a_m$$

Q. Solve the following equation for $y(0.2)$

$$\frac{d^2y}{dx^2} + 2\frac{dy}{dx} - 3y = 6x$$

given $y(0) = 0$, $y'(0) = 1$. Use Heun's method.

Soln:

Given $\frac{d^2y}{dx^2} + 2\frac{dy}{dx} - 3y = 6x$

$$\frac{d^2y}{dx^2} = 6x + 3y - 2\frac{dy}{dx}$$

Let

$$y = y_1, \quad \frac{dy}{dx} = y_2$$

Then

$$f_1 := \frac{dy_1}{dx} = y_2, \quad y_{10} = 0$$

$$f_2 := \boxed{\frac{dy_2}{dx} = 6x + 3y - 2y_2}, \quad y_{20} = 1$$

Let $h = 0.2$

$m_1(1) = y_{20} = f_1(x_0, y_{10}, y_{20}) = f(0, 0, 1) = 1.$

$m_1(2) = f_2(x_0, y_{10}, y_{20}) = f(0, 0, 1) = 6\times0 + 3\times0 - 2\times1 = -2$

$m_2(1) = f_1[x_0 + h, y_{10} + h m_1(1), y_{20} + h m_1(2)]$

$= f_1[0.2, 0 + 1\times0.2, 1 + 0.2\times(-2)] = y_{20} + h m_1(2)$

$= 1 + 0.2\times(-2) = 0.6$

$m_2(2) = 6(0.2) + 3(0 + 0.2\times1) - 2(0.6) = 1.2 + 0.6 - 1.2 = 0.6$

$m(1) = \frac{1 + 0.6}{2} = 0.8, \quad m(2) = \frac{-2 + 0.6}{2} = -0.7$

$\therefore y_1(0.2) = y_1(0) + 0.2\, m(1) = 0 + 0.2\times0.8 = 0.16$

$y_2(0.2) = y_2(0) + 0.2\, m(2) = 1 - 0.2\times0.7 = 0.86$

CH#

* Solution of Partial Differential Equation :-

Any equation with partial derivative is known as partial differential equation. for eg:

i) Laplace equation
ii) Poisson's equation
iii) parabolic equations
iv) hyperbolic equations

* Laplace Eqn :-

An equation in the form of

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0$$

or

$$\nabla^2 f = 0$$

or

$$f_{xx} + f_{yy} = 0$$

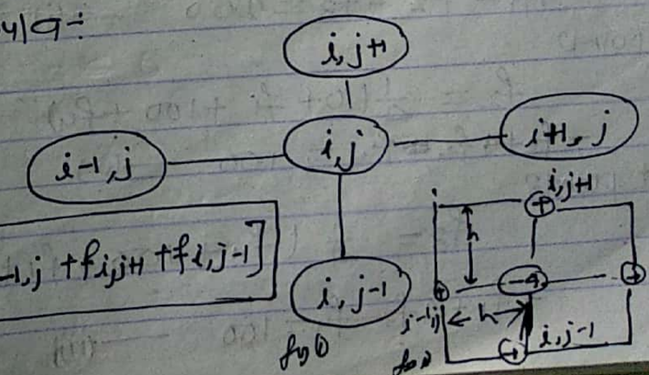
or

$$u_{xx} + u_{yy} = 0$$

is called Laplace partial diff. eqn.

Solution of Laplace equation

Five point formula :-

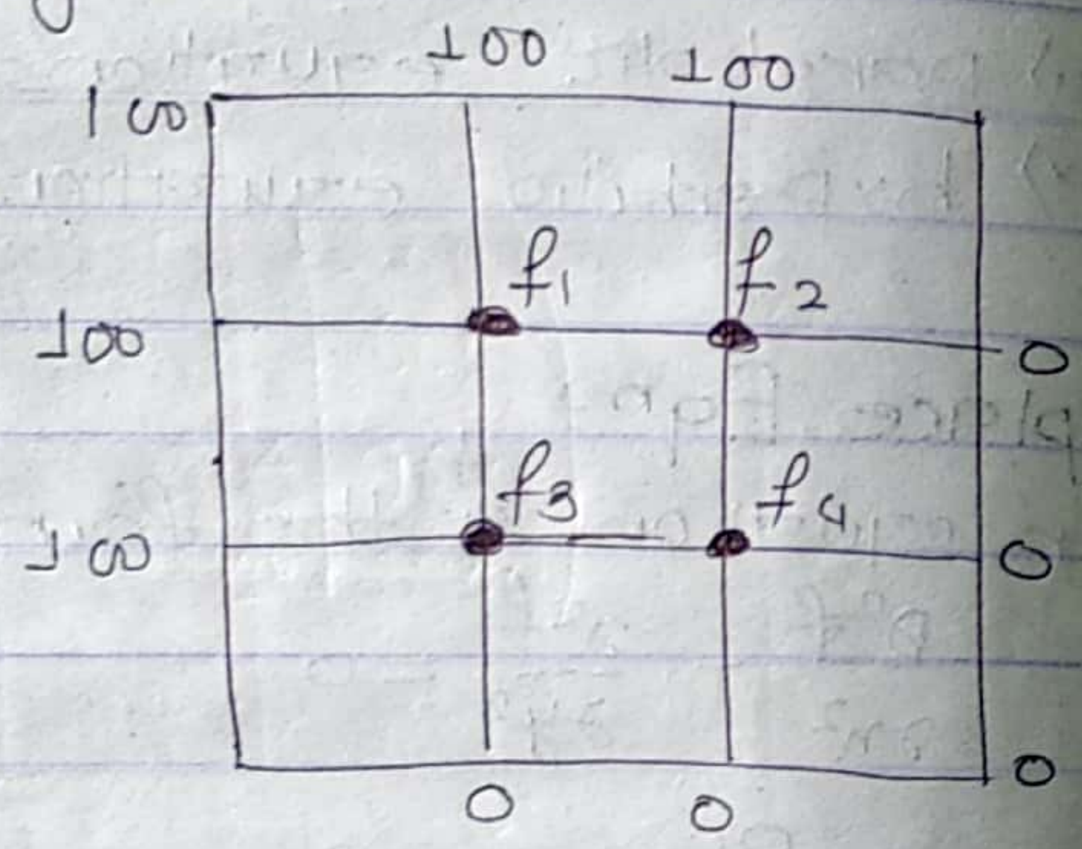


$$\boxed{f_{i,j} = \frac{1}{4}\left[f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1}\right]}$$

Example:-

Consider a steel plate of size 15 cm x 15 cm. If two of the sides are held at 100°C and the other two sides are held at 0°C, what are the steady ~~state~~ temperature at interior points assuming a grid size of 5 cm x 5 cm.

Soln:-



Note:- A problem with the values know on each boundary is said to have Dirichlet boundary conditions.

The system of eqn is as follows

At point 1

$$f_1 = \frac{1}{4}(f_2 + 100 + 100 + f_3)$$

$$4f_1 - f_2 - f_3 = 200 \quad \text{—(i)}$$

At point 2

$$f_2 = \frac{1}{4}(0 + f_1 + 100 + f_4)$$

$$-f_1 + 4f_2 - f_4 = 100 \quad \text{—(ii)}$$

At point 3

$$f_3 = \frac{1}{4}(f_4 + 100 + f_1 + 0)$$

$$-f_1 + 4f_3 - f_4 = 100 \quad \text{—(iii)}$$

Scanned by CamScanner

By $R_3 \to 15R_3 + R_2$ & $R_4 \to 15R_4 + R_2$

$$\sim \left(\begin{array}{cccc|c} 4 & -1 & -1 & 0 & 200 \\ 0 & +15 & -1 & -4 & 600 \\ 0 & 0 & 224 & -64 & 9600 \\ 0 & 0 & -16 & 56 & 600 \end{array}\right)$$

By $R_4 \to 14R_4 + R_3$

$$\sim \left(\begin{array}{cccc|c} 4 & -1 & -1 & 0 & 200 \\ 0 & +15 & -1 & -4 & 600 \\ 0 & 0 & 224 & -64 & 9600 \\ 0 & 0 & 0 & 720 & 18000 \end{array}\right)$$

$\therefore$ $4f_1 - f_2 - f_3 = 200$ — (v)

$+15f_2 - f_3 - 4f_4 = 600$ — (vi)

$224f_3 - 64f_4 = 9600$ — (vii)

$720f_4 = 18000$ — (viii)

from (viii)

$$f_4 = \frac{18000}{720} = 25$$

from (vii)

$$f_3 = \frac{9600 + 64 \times 25}{224} = 50$$

from (vi)

$$f_2 = \frac{+50 + 4 \times 25 + 600}{15} = 50$$

from (v),

$$f_1 = \frac{200 + 50 + 50}{4}$$

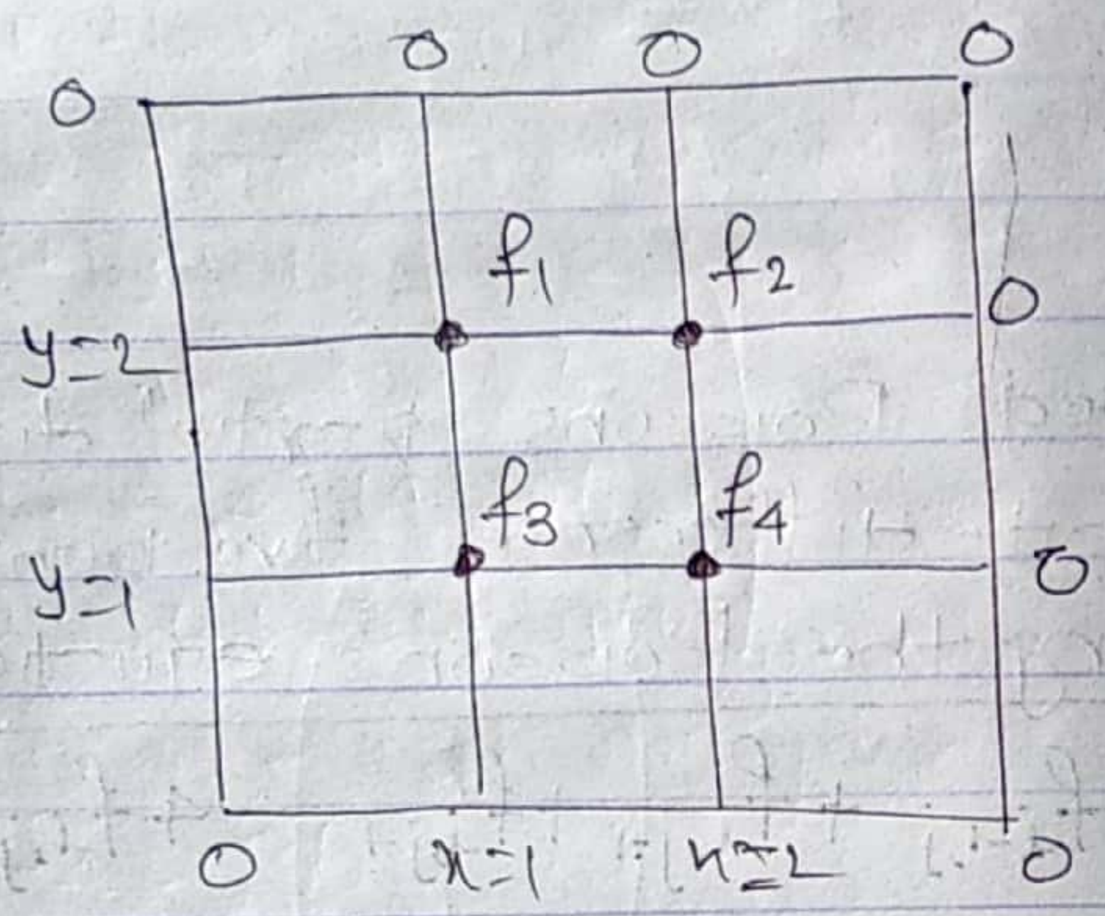
$\therefore f_1 = 75$

Scanned by CamScanner

Q. Solve the poisson equation $\nabla^2 f = 2x^2y^2$ over the square domain $0 \le x \le 3$ and $0 \le y \le 3$ with $f = 0$ on the boundary & $h = 1$

Soln:-

The domain is divided into squares of one unit size as illustrated below:



Applying $f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1} - 4f_{i,j} = h^2 g(x,y)$

at point 1

$$f_2 + 0 + 0 + f_3 - 4f_1 = (1)^2 \cdot 2 (1)^2 (2^2)$$

$$-4f_1 + f_2 + f_3 = 8 \quad \text{—(i)}$$

at point 2

$$0 + f_1 + 0 + f_4 - 4f_2 = 1^2 \cdot 2 \cdot 2^2 \cdot 2^2$$

$$f_1 - 4f_2 + f_4 = 32 \quad \text{—(ii)}$$

at point 3

$$f_4 + 0 + f_1 + 0 - 4f_3 = 1^2 \cdot 2 \cdot 1^2 \cdot 1^2$$

$$f_1 - 4f_3 + f_4 = 2 \quad \text{—(iii)}$$

at point 4,

$$0 + f_2 + f_3 + 0 - 4f_4 = 1^2 \cdot 2 \cdot 2^2 \cdot 1^2$$

$$f_2 + f_3 - 4f_4 = 8 \quad \text{—(iv)}$$

Scanned by CamScanner

By $R_3 \to 15R_3 + R_2$ & $R_4 \to 15R_4 + R_2$

$$\left(\begin{array}{cccc|c} -4 & 1 & 1 & 0 & 8 \\ 0 & -15 & 1 & 4 & 136 \\ 0 & 0 & -224 & 64 & 376 \\ 0 & 0 & 16 & -56 & 256 \end{array}\right)$$

By $R_4 \to 14R_3 + R_3$

$$\left(\begin{array}{cccc|c} -4 & 1 & 1 & 0 & 8 \\ 0 & -15 & 1 & 4 & 136 \\ 0 & 0 & -224 & 64 & 376 \\ 0 & 0 & 0 & -720 & 3960 \end{array}\right)$$

$\therefore \quad -720 f_4 = 3960$

$\therefore f_4 = -5.5$

$-224 f_3 + 64 f_4 = 376$

$\Rightarrow \quad f_3 = \dfrac{64 f_4 - 376}{224} = \dfrac{-64 \times 5.5 - 376}{224}$

$f_3 = -3.24$

$-15 f_2 + f_3 + 4 f_4 = 136$

$\Rightarrow \quad f_2 = \dfrac{f_3 + 4 f_4 - 136}{15}$

$= \dfrac{-3.24 - 5.5 \times 4 - 136}{15}$

$f_2 = 10.79$

$-4 f_1 + f_2 + f_3 = 8$

$\Rightarrow f_1 =$

* Parabolic Equation :→

Parabolic equation for which $b^2 - 4ac = 0$ describe the problems that depend on space and time variables.

A popular case for parabolic type of eqn is the study of heat flow in one-dimensional direction in an insulated rod. Such problems are governed by both boundary and initial conditions.

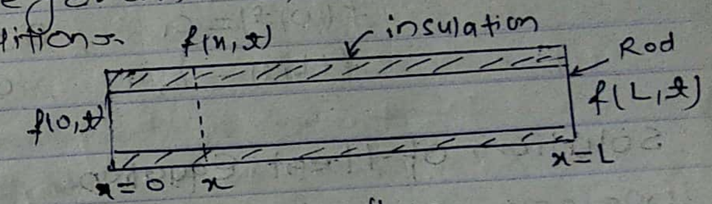


fig :- Heat flow

Let f represents the temprature at any point in rod whose distance from the left end is x. Heat is flowing from left to right under the influence of temprature gradient. The temprature $f(x,t)$ in the rod at the position x and time t, is governed by the heat equation

$$K_1 \frac{\partial^2 f}{\partial x^2} = K_2 K_3 \frac{\partial f}{\partial t} \quad \text{—(i)}$$

where, K_1 = coefficient of thermal conductivity

K_2 = specific heat

K_3 = Density of the material.

Eqn (i) may simplified as

$$\boxed{K f_{xx}(x,t) = f_t(x,t)} \quad \text{—(ii)}$$

where $K = \dfrac{K_1}{K_2 K_3}$

The initial condition will be the initial temprature at all points along the rod.

$$f(x,0) = f(x), \quad 0 \le x \le L$$

The boundary conditions $f(0,t)$ and $f(L,t)$ describe the temprature at each end of the as function of time. If they are held constant, then

$$f(0,t) = C_1, \quad 0 \le t < \infty$$
$$f(L,t) = C_2, \quad 0 \le t < \infty$$

Solution of Heat Equation

We can solve the heat eqn given by eqn (ii) using the finite difference formula given below:

$$f_t(x,t) = \frac{f(x, t+\tau) - f(x,t)}{\tau} = \frac{1}{\tau}(f_{i,j+1} - f_{i,j}) \quad \text{— (ii)}$$

$$f_{xx}(x,t) = \frac{f(x-h,t) - 2f(x,t) + f(x+h,t)}{h^2} \quad \text{— (iii)}$$

$$= \frac{1}{h^2}(f_{i-1,j} - 2f_{i,j} + f_{i+1,j})$$

solving (ii), (iii) & (iv) we get

$$\tau \le \frac{h^2}{2K} \quad \text{—— (x)}$$

&

$$f_{i,j+1} = \frac{1}{2}[f_{i+1,j} + f_{i-1,j}] \quad \text{—— (xx)}$$

where $\tau = \Delta t$
$h = \Delta x$

Q. Solve the equation

$$2f_{xx}(x,t) = f_t(x,t), \quad 0 < t < 1.5 \;\&\; 0 < x < 4$$

given the initial condition

$$f(x,0) = 50(4-x), \quad 0 \le x \le 4$$

and the boundary condition

$$f(0,t) = 0, \quad 0 \le t \le 1.5$$
$$f(4,t) = 0, \quad 0 \le t \le 1.5$$

Soln:

Given,

$$2f_{xx}(x,t) = f_t(x,t) \quad \text{— (i)}$$

comparing with

$$K f_{xx}(x,t) = f_t(x,t)$$

$$K = 2$$

Assume $\Delta x = h = 1$

We know,

$$\tau \le \frac{h^2}{2K}$$

$$\tau \le \frac{1^2}{2\times 2}$$

$$\therefore \tau \le 0.25$$

Taking $\tau = 0.25$

We have $f_{i,j+1} = \frac{1}{2}(f_{i-1,j} + f_{i+1,j})$

t ↓ x →	0	1	2	3	4
0	0	150	100	50	0
0.25	0	50	100	50	0
0.50	0	50	50	50	0
0.75	0	25	50	25	0
1.00	0	25	25	25	0
1.25	0	12.5	25	12.5	0
1.50	0	12.5	12.5	12.5	0

$f(x,0) = 50(4-x)$

* Solution of hyperbolic Equation:

Wave Eqn:-

The general displacement of string f varies with the time t and displacement x along the string. The displacement $f(x,t)$ is governed by wave equation

$$T\frac{\partial^2 f}{\partial x^2} = \rho\frac{\partial^2 f}{\partial t^2} \quad \text{—(1)}$$

where

T = tension in string

ρ = mass per unit length

Eqn(i) can be simplified as

$$T f_{xx}(x,t) = \rho f_{tt}(x,t)$$

The boundary conditions for the vibrating string problem are

$$f(0,t) = 0 \quad ; \quad 0 \le t \le b$$
$$f(L,t) = 0 \quad , \quad 0 \le t \le b$$

Two initial conditions are

$$f(x,0) = f(x) \quad , \quad 0 \le x \le a$$
$$f_t(x,0) = g(x) \quad , \quad 0 \le x \le a$$

Solution of wave eqn is $\boxed{1 - \frac{T\tau^2}{\rho h^2} = 0}$ —(1)

$$\boxed{f_{i,1} = \frac{1}{2}\left(f_{i+1,0} + f_{i-1,0}\right)} \quad \text{—(iii)}$$

$$\boxed{f_{i,j+1} = f_{i+1,j} + f_{i-1,j} - f_{i,j-1}} \quad \text{—(iv)}$$

Scanned by CamScanner

* Fitting Of Linear Curve

Algorithm of Linear Regression

1. Read data values
2. Compute sum of powers and products
   $\Sigma x_i, \Sigma y_i, \Sigma x_i^2, \Sigma x_i y_i$
3. Check whether the denominator of the eqn for b is zero.
4. Compute b and a
5. Print out the equation.
6. Interpolate data, if required.

Program

```
#include<math.h>
#define MAX 10
#define EPS 0.000001
main()
{
 int i,n;
 float x[10], y[10];
 float sumx, sumy, sumxx, sumxy, xmean,
    ymean, denom, a,b;
 printf("\n Linear Regression \n\n");

 printf("\n Input number of data points, n\n");
 scanf("%d", &n);
 printf(" \n Input x and y values \n");
 printf(" One set on each line\n");
  for(i=1; i<=n; i++)
     scanf("%f %f", &x[i], &y[i]);
```

```
sumx = 0.0;
sumy = 0.0;
sumxx = 0.0;
sumxy = 0.0;
for (i=1; i<=n; i++)
{
    sumx = sumx + x[i];
    sumy = sumy + y[i];
    sumxx = sumxx + x[i]*x[i];
    sumxy = sumxy + x[i]* y[i];
}

xmean = sumx/n;
ymean = sumy/n;
denom = n*sumxx - sumx * sumx;
if(fabs(denom) > EPS)
{ b = (n*sumxy - sumx * sumy)/denom;
  a = ymean - b* xmean;
printf("\n Linear Regression Line y = a+bx\n");
printf("\n The coefficients are :\n");
printf("a = %f \n", a);
printf("b = %f\n", b);
}
else
  {
    printf("\n No solution\n");
  }
getch()
}
```

Scanned by CamScanner

CHAPTER#01

INTRODUCTION TO NUMERICAL COMPUTING

Numerical computing is an approach for solving complex mathematical problems using only simple arithmetic operations.

* PRocess Of Numerical Computing:

The process of numerical computing can be roughly divided into the following four phases:

1. formulation of a mathematical model
2. construction of an appropriate numerical method.
3. implementation of the method to obtain a solution.
4. validation of the solution.

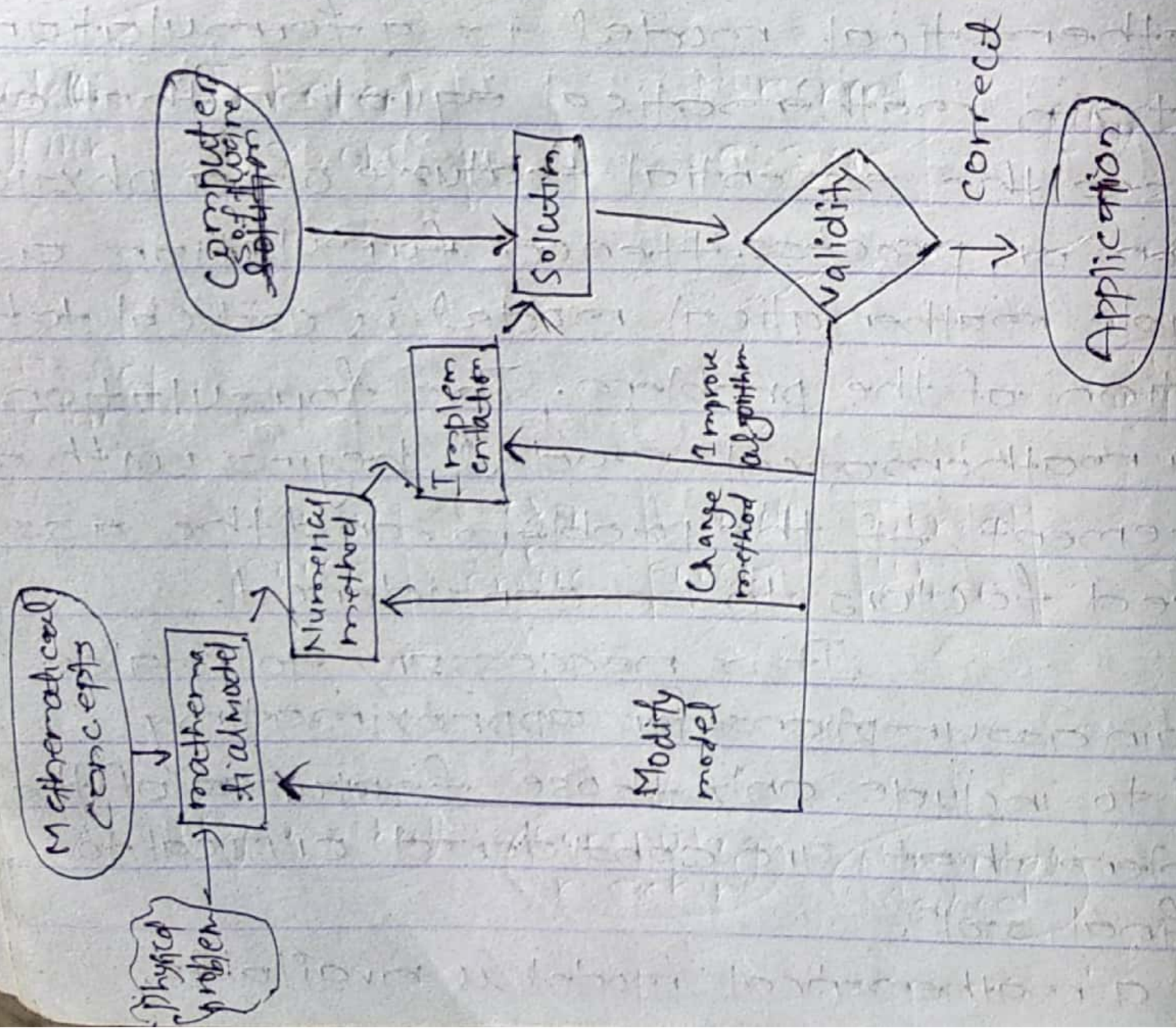
A mathematical model is a formulation of certain mathematical equation that expresses the essential feature of a physical system or process. Hence, formulation a suitable mathematical model is critical to the solution of the problem. The formulation of a mathematical model begins with a statement of the problem and the associated factors to be considered.

It is neccessary to make certain assumptions for approximating and to include only those features of the problem that are considered critical to the final soln.

Once a mathematical model is available,

our first step would be try to obtain an explicit analytical solution. In most cases, the mathematical models may not be amenable to analytical soln or they may not be solved efficiently using analytical techniques.

For a given problem, there might be several alternative numericals method. We must consider different factors or trade-offs before selecting a particular method - such as type of execution, type of computer available, accuracy, speed of execution, and programming and maintenance effort required.



interfaces. They also enable the extension (or reuse) of the existing code without modifying it.

* Taxonomy Of Errors In Numerical Method:

A number of different type of errors arise during the process of numerical computing. All these errors contribute to the total error in the final result. A taxonomy of errors encountered in a numerical process is

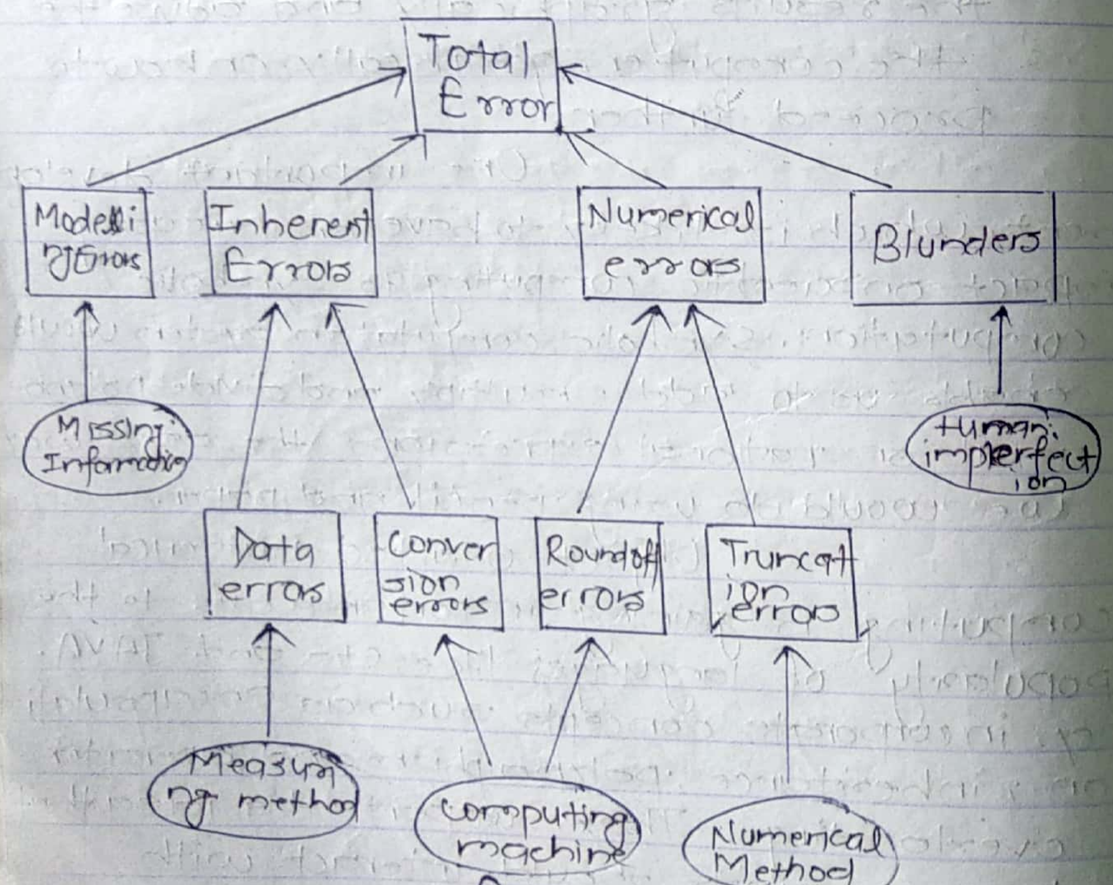


fig: Taxonomy of errors.

* Numerical Errors:

Numerical errors (also known as procedural errors) are introduced during the process of implementation of a numerical method. They come in following two forms

(i) Roundoff errors
(ii) Truncation errors

The total numerical error is the summation of these two errors.

* Absolute and Relative Errors:

Regardless of its source, an error is usually quantified in two different but related ways. One is known as absolute errors and the other is called relative error.

Let us consider that the truth value of data item is denoted by x_t and its approximate value is denoted by x_a. Then, they are related as follows:

Truth value x_t = Approximate value x_a + Error

The error is then given by,

$$\text{Error} = x_t - x_a$$

The error may be negative or positive but what is important is the magnitude of the error and not the sign and, therefore, we normally consider what is known as absolute error which is denoted by

$$e_a = |x_t - x_a|$$

Relative error is nothing but the "normalized" absolute error. The relative error is defined as follows:

$$e_r = \frac{\text{absolute error}}{|\text{truth value}|}$$

$$= \frac{|x_t - x_a|}{|x_t|} = \left|1 - \frac{x_a}{x_t}\right|.$$

More often, the quantity that is known to us is x_a and, therefore, we can modify

```
if (fm == 0)
{
    printf ("\n The required root is %f", xm);
    goto a2;
}
else if (f1 * fm < 0)
{
    x2 = xm;
    f2 = fm;
else
{
    x1 = xm;
    f1 = fm;
}
if (fabs (x2 - x1) < E)
}
printf ("\n The required root is %f", (x1+x2)/2)
goto a2;
}
else
{
goto a3;
}
a2 :
getch ();
}
```

Algorithm for Bisection method

1. Decide the initial values for x_1, x_2 and stopping criterion E
2. Compute $f_1 = f(x_1)$ and $f_2 = f(x_2)$
3. If $f_1 * f_2 > 0$, then $x_1 + x_2$ do not bracket any root and go to step 7. Otherwise continue.
4. Compute $x_m = (x_1 + x_2)/2$ and compute $f_0 = f(x_m)$
5. If $f_1 * f_0 < 0$ then
   set $x_2 = x_m$
   else
   set $x_1 = x_m$
6. If absolute value of $(x_2 - x_1)/x_2$ is less than error E then
   root $= (x_1 + x_2)/2$
   write the value of root
   go to step 7
   else,
   go to step 4
7. stop

```
//Program To evaluate polynomial Using Horners
  Method or
  Program To evaluate f(x) at x

#include<stdio.h>
#include<conio.h>
void main()
{
  int n, i;
  float x, a[20]
  clrscr();
printf("\n Enter the degree of polynomial\n");
  scanf("%d", &n);
printf("Enter the value of x for f(x)\n");
  scanf("%f", &x);
for (i=0; i<=n; i++)
{
  printf("\nEnter the value of a[%d]=", i);
  scanf("%f", &a[i]);
}
  Pn = a[n];
for (i=n-1; i>=0; i--)
  {
    P[i] = P[i+1] * x + a[i];
  }
printf("\n The value of f(x)=%f", P);
  getch();
}
```

Algorithm

$$P_n = a_n$$
$$P_{n-1} = P_n x + a_{n-1}$$
$$P_j = P_{j+1} + a_j$$
$$P_1 = P_2 x + a_1$$
$$f(x) = P_0 = P_1 x + a_0$$

- Evaluate $f(x) = x^3 - 4x^2 + x + 6$, at $x = 2$

$a_3 = 1$, $a_2 = -4$, $a_1 = 1$, $a_0 = 6$

$P_3 = a_3 = 1$

$P_2 = 1 \times 2 + (-4) = -2$

$P_1 = (-2) \times 2 + 1 = -3$

$P_0 = (-3) \times 2 + 6 = 0$ (Answer)

Scanned by CamScanner

// Newton's Raphson Method:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#define f(x) x*x -4*x-10
#define fd(x) 2*x -4
#define E 0.0005
void main()
{
 clrscr();
 float x0,x1,f0,fd0,e;
 printf("\t\t\t --- Newton-Raphson Method---");
 printf("\t\t\t Finding Root of x^2 - 4x-10");
 printf("\n Enter initial value:");
 scanf("%f", &x0);
 printf("\n f(x0)\t\t fd(x0)\t\t x1\t\t Error");
 printf("\n ---------- ");
 begin:
        f0=f(x0);
        fd0 = fd(x0);
        x1 = x0- (f0/fd0);
         e = fabs((x1-x0)/x1);

          if(e<E){
          printf("\n\n Approximate root=%5", x1);
          }
        else {
        printf("\n %2f\t\t %3f\t\t %3f\t\t %4f"
                                  f(x0),fd(x0),x1,e);
            x0=x1
            goto begin
```

// Program for Gauss Jordan Elimination Method

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
 void main()
 {
  int i,j,k;
 float a[10][10],x,r;
  clrscr();
 printf("\n Enter the no. of unknown:");
  scanf("%d", &n);
 printf(" Enter the elements of the augumenta-
        ed matrix [a|b]");
 for(i=0;i<=n;i++)
 {
   for(j=1;j<=n;j++)
   {
   printf("\n Enter the element[%d][%d]=",i,j);
    scanf("%f",&a[i][j]);
   }
 }
 printf("\n Enter the augumented matrix");
  for(i=1;i<=n;i++)
  {
    printf("\n");
    for(j=1;j<=n;j++)
    {
      printf("%2f\t", a[i][j]);
    }
```

Scanned by CamScanner

```
//row operation
 for(i= 1;i<=n:i++)
 {
   for (j=1;j<=n;j++)
   {
     if (j!=i)
     r = a[j][i]/a[i][i];
      for (k=1;k<=n+1;k++)
     a[j][k]-= r*a[i][k];
   }}}
 printf("\n The row operated matrix:");
 for (j=1;j<=n;i++)
  {
    printf("\n")
     for (j=1;j<=n;j++)
    }
     printf("%2f\t", a[i][j]);
   }}
 // The solution calculation begins from here
for (i=1;i<=n;i++)
  {
    x = a[i][n+1]/a[i][i];
   printf("\n the value of %d =%2f ",i,x);
  }
   getch();
 }
```

* Algorithm for Gauss-Jordan Elimination

1. Normalize the first row by dividing it by its pivot element.
2. Eliminate x_1 from all other equations
3. Normalize the 2nd equation by dividing by its pivot element.
4. Eliminate x from all equation above and below the normalized pivotal equation.
5. Repeat this process until x_n is eliminated from all but not from last equation.
6. The result b vector is the soln vector.

```
// Solution by Gauss elimination Method//
#include <stdio.h>
#include <conio.h>
void main()
 int n,i,j,dum;
 float a[10][10], k, sum, x[10];
 clrscr();
printf("\n Enter the no. of eqn or unknown=");
 scanf("%d", &n);
printf("\n Enter the element of augumented
      matrix ");
 for (i=1;i<=n;i++)
  {
    for (j=1;j<=n;j++)
  }
  printf("\n Enter the element a[%d][%d]=",
                                  i,j);
   scanf("%f", &a[i][j]);
  }}
printf("\n The Enter augumented matrix is");
 for (i=1;i<=n;i++)
 }
  printf("%n");
   for (j=1;j<=n;j++)
   }
    printf("%0.2f\t", a[i][j]);
   }}
// Triangularisation begins from here
 for( dum=1; dum<=n-1; dum++)
 }
  ~~for (j=dum;i<=n;i++)~~
   for( i=dum; i<=n-1 ; i++)
   {
   k = a[i+1][dum] / a[dum][dum];
   for (j=dum;j<=n+1;j++)
    }
     a[i+1][j] -= (k* a[dum][j]);
   }}}
 printf("\n The triangularised augumented matrix");
  for( i=1; i<=n;i++)
   {
    printf("\n");
    for (j=1;j<=n;j++)
    }
    printf("%n%.2f\t", a[i][j]);
   }}
// Backward substitution begins from here
  for (i=n; i>0; i--)
   {
    sum=0;
   for (j=n;j>0;j--)
   }
    (j>i)
   }
   sum+ =(a[i][j] *x[j]);
   }
    else
      }
```

```
x[i] = (a[i][n+1] - sum)/a[i][i];
break;
}}}
// The output begins from here
printf("\n The calculation of given system of
   linear equation is:");
for (i=1; i<=n; i++)
{
  printf("\n x[%d] = %2f", i, x[i]);
}
  getch();
}
```

```
// Solution Of Differential Equation by Euler's Method//
#include<stdio.h>
#include<conio.h>
#include<math.h>
#define f(x,y) (3+x*x+y)
void main()
{
  int i,n;
  float x0,y0, h,x, y[10];
   clrscr();
 printf("\n Enter the initial value of x & y :");
  scanf("%f %f", &x0, &y0);
  printf("Enter the value of x where we have
     to find y(x) :");
     scanf("%f", &x);
     n = (x-x0)/h;
     printf("\n No. of calculation = %d", n);
     for(i=1; i<=n; i++)
    {
     y[i] = y0 + f(x0, y0)*h;
     x0 = x0+h;
     y0 = y[i];
    }
  printf("\n y(x) at x is = %f", y0);
    getch();
  }
```

// Program To Solve the R.K -4 Method

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#define f(x,y) 2*y/x
#define h 0.25
void main()
{
  int i,n;
  float x,x0,y0,m1,m2,m3,m4,y[10];
  clrscr();
printf("\n Enter the initial value of x & y");
  scanf("%f %f", &x0, &y0);
printf("Enter the value of x where you have to find y(x)");
  scanf("%f", &x);
  n=(x-x0)/h);
printf("%d", n);
for(i=1;i<=n;i++)
{
  m1 = f(x0,y0);
  m2 = f(x0+h/2, y0+h/2*m1);
  m3 = f(x0+h/2, y0+m2*h/2);
  m4 = f(x0+h, y0+m3*h);
  y[i] = y0+(m1+2*m2+2*m3+m4)*h/6;
  x0 = x0+h;
  y0 = y[i];
}
printf("\n The value of f(x) at x=%f", y[i]);
  getch();
}
```

Algorithm

```
// Program To find Integral Using Trapezoidal Rule
#include <stdio.h>
#include <conio.h>
#include <math.h>
#define f(x) 1/(1+x*x)
void main()
{
int i,n;
float x0,xn, h,s,s1, x[30], f[30];
clrscr();
printf("\n Enter the lower and upper limit:");
scanf("%f%f", &x0, &xn);
printf("\n Enter the no. of strips: ");
scanf("%d", &n);

h= (xn-x0)/n;

for( j=0; j<=n ; j++)
{
x[j] = x0;
f[j] = f(x0);
x0 = x0+h;
}
s1=0.0;
s1= f(x0)+f(xn);
for(j=1;j<n ; j++)
{
s = s1 + 2* f(x0+j*h);
}
s = h/2 * s;
```

```
printf("\n Integral is =%f", s);
getch();
}

// Program To find Integral by Using Simpson's
1/3 Rule //

#include<stdio.h>
#include<conio.h>
#define f(x)x
void main()
{
int n,i;
float a,b,h,s,s1, I, x[30], f[30];
clrscr();
printf("\n The lower and upper limits:");
scanf("%f%f", &a, &b);
printf("\n Enter the no. of strips:");
scanf("%d", &n);
h= (b-a)/n;
for (i=1;i<=n;i++)
{
x[i]=a;
f[i]=f(a);
a= a+h;
}
s1= 0.0;
for(j=1; i<=n,i+
```

```
// Program To find Integral Using Trapezoidal Rule
#include <stdio.h>
#include <conio.h>
#include <math.h>
#define f(x)  1/(1+x*x)
 void main()
{
 int i,n;
   float x0,xn, h, s, s1, x[30], f[30];
   clrscr();
   printf("\n Enter the lower and upper limit:");
    scanf("%f%f", &x0, &xn);
   printf("\n Enter the no. of strips:");
    scanf("%d", &n);
   h= (xn-x0)/n;
   for( i=0; i<=n; i++)
   {
    x[i] = x0;
    f[i] = f(x0);
    x0 = x0+h;
   }
    s1=0.0;
    s1= f(x0)+f(xn);
   for(i=1; i<n; i++)
   {
    s = s1 + 2*f(x0+i*h);
   }
    s = h/2 * s;
```

```
   printf("\n Integral is =%f", s);
    getch();
   }

// Program To find Integral by Using Simpson's
    1/3 Rule //

#include<stdio.h>
#include<conio.h>
#define f(x) x
 void main()
 {
  int n,i;
  float a,b,h,s,s1, I, x[30], f[30];
   clrscr();
  printf("\n The lower and upper limits:");
    scanf("%f%f", &a, &b);
  printf("\n Enter the no. of strips:");
    scanf("%d", &n);
   h= (b-a)/n;
    for (i=1; i<=n; i++)
    {
      x[i]= a;
      f[i] = f(a);
      a= a+h;
    }
     s= 0.0;
   for(i=1; i<=n-1, i+2)
```

```
{
  s+ = f[i];
  }.s1= 0.0;
 for (i=2 ; i<= n-1 ; i+2)
 {
    s1 + = f[i];
  }
  I = h/3 * ( f[0]+f[n] + 4*s +2*s1);
 printf("\n Integral value is = %f", I);
 getch();
}
```

Algorithm:

1. Read the lower and upper limits of integral a & b;
2. Read the divided no. of strip(i.e. no. of sub interwals, n;
3. Calculate step size, h= (b-a)/n;
4. Now evaluate f(x) at each increment h i.e. f(xi) & xi = a+ib, i = 0,1, .. n
5. Now apply composite Simson's 1/3 rule formula
6. I gives the integral value.
7. stop.

// Program To find Integration by Using Simson's 3/8 Rule:

```
#include <stdio.h>
#include <conio.h>
#include<math.h>
#define f(x) 1/x
void main()
{
  float a,b, h, s, s1, I, x[30], f[30];
  int i,n;
   clrscr();
  printf("\n Enter lower & upper limits:");
   scanf("%f %f", &a, &b);
   printf(" Enter the no. of strips");
   scanf("%d", &n);
   h= (b-a)/n;
  printf(" step size = %f", h);
   for (i = 0; i <n; i++)
   {
     x[i] = a;
     f[i] = f(a);
     a+ = h;
   }
     s = 0.0;
   for(i = 3 ; i <= n-1 ; i = i+3)
   {
     s = s+f[i];
   }
     s1 = 0.0;
   for( i = 1 ; i <= n-1 ; i++)
```

```
{
  if (i%3 != 0)
    s1 = s1 + f[i];
}

  I = (3*h (f[0] + f[n] + 2.0*s + 3.0*s1))/8;
printf("\n Integral value is = %f", I);
 getch();
}
```

```
//Program for Lagrange Interpolation//
#include <stdio.h>
#include <conio.h>
 void main()
 {
  clrscr()
 float x0, x[300], f[300], sum=0.0, n;
 for m=1.0;
 printf("\n Input no. of data points");
  scanf("%f", &n);
 for (int i=1; i<=n; i++)
  {
    printf("\n Enter the data and value
    of function at that point");
    scanf("%f %f", &x[i], &f[i]);
  }
 printf("\n Input x at which interpolation
        is required");
  scanf("%f", &x0);
 for (i=1; i<=n; i++)
  {
   m=0.1;
  for (int j=1; j<=n; j++)
   {
    if (i!=j)
     m = m*(x0-x[j])/(x[i]-x[j]);
   }
   sum = sum+m*f[i];
  }
printf("\n%f", sum); getch(); }
```

```
// Newton's Interpolation
#include <stdio.h>
#include<conio.h>
# include <math.h>
 void main()
 {
  int i,j,n;
  float xp, fp, sum, Pi, x[10], f[10], a[0], d[10][0];
  printf("\n Input no. of data points \n");
   scanf("%d", &n);
  printf("\n Input values of x and f(x) ;\n");
  printf(" One set on each line \n");
   for(i = 1;i<=n ;i++)
      scanf("%f %f", &x[i], &f[i]);
 /* constructing difference table */
      for (i=1 ;i<=n;i++)
          d[ i][1] = f[i]
          for (j=2 ;j <=n;j++)
           for(i=1; i<=n-j+1 ;i++)
           d[i][j] = (d[i+1][j-1] - d[i][j-1])/
                                  (x[i+j-1] - x[i])
 /* Set the coefficients of interpolation polynomial*/
          for (j=1 ;j<=n ;j++)
           a[j]=d[1][j];
 /* compute interpolation value */
   printf("\n Input xp where interpolation is required")
      scanf("%f", &xp);
       sum = a[1];
```

```
for (i= 2;i<=n ;i++)
 }
     Pi =1.0;
     for (j=1 ;j <=i-1 ;j++)
          Pi = Pi * (xp-x[j]);
       sum = sum+a[i]* Pi ;
     }
        fp=sum;
 /* write results */
  printf("\n");
 printf(" Newton's Interpolation \n");
  printf("\n ");
  printf("\n Interpolated value\n");
  printf("at x = %f is %f \n", xp, fp);
 }
 /* End of main program */
```

HEUN (A first Order differential eqn $y' = f(x,y)$ using Heun's method)

```
void main()
{
  int i,n;
  float x,y,xp,h,m1,m2;
  float func(float,float);
  printf("\n Solution By Heun's Method\n\n");
/* Reading initial data */
  printf("Input initial values of x and y\n");
  scanf("%f", &x, &y);
  printf("Input x at which y is required\n");
  scanf("%f", &xp);
  printf("\n Input step-size, h\n");
  scanf("%f",&h);
/* computing no. of steps required */
    n = (int)((xp-x)/h+0.5);
/* computing y recursively at each step */
  for(i=1;i<=n;i++)
  {
    m1 = func(x,y);
    m2 = func(x+h, y+m1*h);
    x = x+h;
    y = y+0.5*h*(m1+m2);
  printf("%5d %10.6f %10.6f\n", i, x, y);
  }
/* write the final result */
  printf("\n value of y at x=%f is %f\n", x,y);
}
/* End of main() program */
/* Defining subroutine func() */
  float func (float x, float y)
  {
    float f;
    f = 2.0 * y/x;
    return(f);
  }
/* End of routine func() */
/* - - - - - - - - */
```