Pimpri Chinchwad Education Trust's

Pimpri Chinchwad College of Engineering, Pune

Department of Mechanical Engineering

Continuous Assessment Sheet

Course Name: Numerical Methods and Optimization (BME6413) **Class:** T.Y. B. Tech Mechanical **Academic Year:** 2022-23

Semester: VI

Name of Student: Sarthak . Akshay. Patil **Class:** T.Y **Div.:** B **Roll No.:** TYMEB228

Sr. No.	Title of Assignment/ Experiment	Start Date	Completion Date	Knowledge (10)	Skill (10)	Attitude (5)	Total 25
1	Introduction to MATLAB		13/2/24				
2	Roots of Equation		13/2/24				
3	Simultaneous Equations		21/2/24				15
4	Numerical Integration		21/2/24				
5	Curve Fitting	2/4	16/2/24				
6	Interpolation		16/2/24				
7	Ordinary differential Equation		16/2/24				
8	Laplace		16/2/24				
9	Matlab Solver		16/2/24				
			Total =				

Signature of Student

Signature of Course Teacher

PIMPRI CHINCHWAD EDUCATION TRUST'S

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

Sector No. 26, Pradhikaran, Nigdi, Pune - 411044

Department Of Mechanical Engineering

Year and Course TY NMO Lab Roll No.: TYMEB228

INDEX

Sr. No.	Title	Page No.	Date	Signature	Remarks
1)	Introduction to MATLAB		1-1-24		
2)	Roots of Equations		9-1-24		
	1) Bisection method				
	2) Newton Raphson method				
3)	Simultaneous Equation		16-1-24		
	1) Gauss Elimination				
	2) Gauss Seidel				
	3) TDMA				
4)	Numerical Integration		6-2-24		
	1) Trapezoidal method				
	2) Simpsons 1/3 method				
	3) Simpsons 3/8 method.				

This is certify that Shri. Sarthak Abhay. Patil has carried out the above mentioned Numerical method 4 optimization Lab Practicals / Term Work in the Mechanical Department of Pimpri Chinchwad College of Engineering, Pune-44.

Date: 16/04/24

Subject Incharge Head of Department

Pimpri Chinchwad College of Engineering, Pune - 44

PIMPRI CHINCHWAD EDUCATION TRUST'S

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

Sector No. 26, Pradhikaran, Nigdi, Pune - 411044

Department Of Mechanical Engineering

Year and Course TY NMO Lab Roll No. : TYMEB228

INDEX

Sr. No.	Title	Page No.	Date	Signature	Remarks
5)	Curve fitting		13-2-24		
	1) Straight line				
	2) Quadratic equation				
	3) Power curves				
6)	Interpolation		27/2/24		
	1) Lagranges method				
	2) Newtons forward diff				
7)	Ordinary differential Equa Equation		19-3-24		
	1) Eulers				
	2) modified Eulers				
	3) RK2 & RK4				
	4) simultaneous RK-2				
8)	Laplace Equation		26-3-24		
9)	matlab solver		2-4-24		

This is certify that Shri. Sarthak . Abhay . Patil has carried out the above mentioned Numerical method & optimization lab Practicals / Term Work in the Mechanical Department of Pimpri Chinchwad College of Engineering, Pune-44.

Date : 16/04/24

Subject Incharge Head of Department

Pimpri Chinchwad College of Engineering, Pune - 44

Assignment no 1

Introduction to Matlab:-

1) What is Matlab:-

-> Matlab is a proprietary multiparadigm programming language & numeric computing environment developed by Mathworks. Matlab allows matrix manipulations, plotting of functions & data, implementation of algorithms, creation of user interfaces & interfacing with programs written in other languages.

2) Commands in Matlab:-

a) Commands for Managing the session:-

Commands	Purpose
clc	Clear command window
clear	Remove variables from memory
exist	Check for existence of file
global	Declares variables to be global
help	Search for a helth topic
lookfor	Searches help enteries keyword
quit	Stop matlab
who	Lists current variables
whos	Lists current variables (long displays)

b) Commands for working with system:-

Commands	Purpose
cd	Changes current display
data	Display current data
delete	Deletes a file
diary	Switches on
dir	Lists all files in current directory
load	Loads workspace variables from file
path	Display search path
pwd	Display current directory
save	Saves workspace variables in a file
type	Display content of a file.
what	Lists all matlab file in current director
wk1read	Reads .WK1 spreadsheet file.

c) Input & Output Commands:-

Commands.	Purpose
disp	Display contents of an array/string
fscanf	Read formatted data from a file
format	Controls screen-display format
fprint	Performs formatted writes to file
input	Disp prompts & wait for input
;	suppresses screen printing.

d) Vector, matrix & array commands:-

Commands	Purpose
cat	Concatenates arrays
find	Finds indices of nonzero elements
linspace	Creates regularly spaced vector
logspace	Creates logarithmically spaced vectors
prod	Product of each column
sort	Sorts each column
size	Computes array size
sum	Sums each column
dot	Computes Matrix dot products
det	Computes determinant of an array.

e) Plotting Commands:-

Commands	Purpose
axis	Sets axis limits
fplot	Intelligent plotting of functions
grid	Display gridlines
plot	Generates xy plot
print	Prints plot or saves plot to a file
xlabel	add text label to X-axis
ylabel	add text label to Y-axis
gtext	Enables label placement by mouse.
legend	length placement by mouse.

3) Matlab symbols :-

Symbols	Role
+	Addition
+	Unary plus
−	Subtraction
.*	Element wise multiplication
*	Matrix Multiplication
./	Element wise right division
/	Matrix right division
.\	Element wise left division
\	Matrix left division
.^	Element wise power
^	Matrix power
.'	Transpose
'	Complex conjugated Transpose
==	Equal to
~=	Not Equal to
>	Greater than
>=	Greater than or equal
<	Less than
<=	Less than or equal
&	Find logical AND
|	Find logical OR
&&	Find logical AND (with short circuiting)
||	Find logical OR (with short circuiting)
~	Find logical NOT
,	Comma
: | ;	Colon / semi-colon

4) Inbuilt functions:-

length	Computes no of elements
size	Compute array size
max or min	Returns largest / smallests element
sum	Sums each column
prod	product of each column
abs	absolute value
mean	Return the mean over all element
std	Return the standard deviation
cumsum	Return the cumulative sum
round	Specify the type of rounding.
ceil	rounds values to nearest integer toward positive infinity.
floor	rounds values to nearest integer less than equal to element.

5) Branches, loop & flow control:

1) If else statement:
The if statement evaluates a logical expression & executes a group of statements when the expression is true. The optional else if & else keyword provide for the execution of alternative groups of statement. An end keyword, which matches the if, terminates the last group of statements. The group of statements are delineated by the four keywords - no braces or brackets are involved.

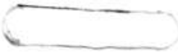
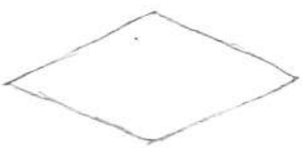
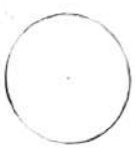
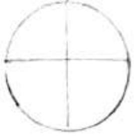
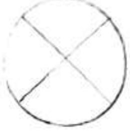
2) Switch & case - statement:
The switch statement executes groups of statements based on the values of a variables or expression. The keywords case & otherwise delineate the groups only the first matching case is executed. There must be an end to match the switch.

3) For loop:
The for loop repeats a group of statement a fixed, predetermined no of times. A matching end delineates statement.

4) While loop: The while loop repeates a group of statements an indefinite no of times under control of logical condition. A matching end delineates the statement.

PCCOE

6) Flowchart symbols :-

Symbols	Name	Function
	Start/End	An oval representation of start & end.
	Arrow	A line is connector between the representative shape that shows relationship
	Input/Output	A parallelogram represents input or output.
	Decision	A diamond indicates a decision
	Process	A rectangle shape indicates process.
	Connector	Indicates an inspection point.
	OR	Logical OR
	Summing Junction	Logical AND

Assignment no: 2

Bisection method

Q) Find a root of eqn $x^3 - 4x - 9 = 0$ using bisection method with accuracy 0.001

→ Let

$$y = f(x) = x^3 - 4x - 9$$

assume

$x = 2$ then $y = f(x) = -4$

$x = 3$ then $y = f(x) = 6$

1st iteration

$x_1 = 2$ $\quad f(x_1) = -4$

$x_2 = 3$ $\quad f(x_2) = 6$

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

$$= \frac{2+3}{2} = 2.5$$

$$f(x_3) = -3.375$$

$$(x_2 - x_1) = 1 > 0.001$$

2nd iteration

$x_1 = 2.5$ $\quad f(x_1) = -3.375$

$x_2 = 3$ $\quad f(x_2) = 6$

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

$$x_3 = \frac{2.5 + 3}{2} = 2.75$$

$$f(x_3) = 0.7698 > 0.001$$

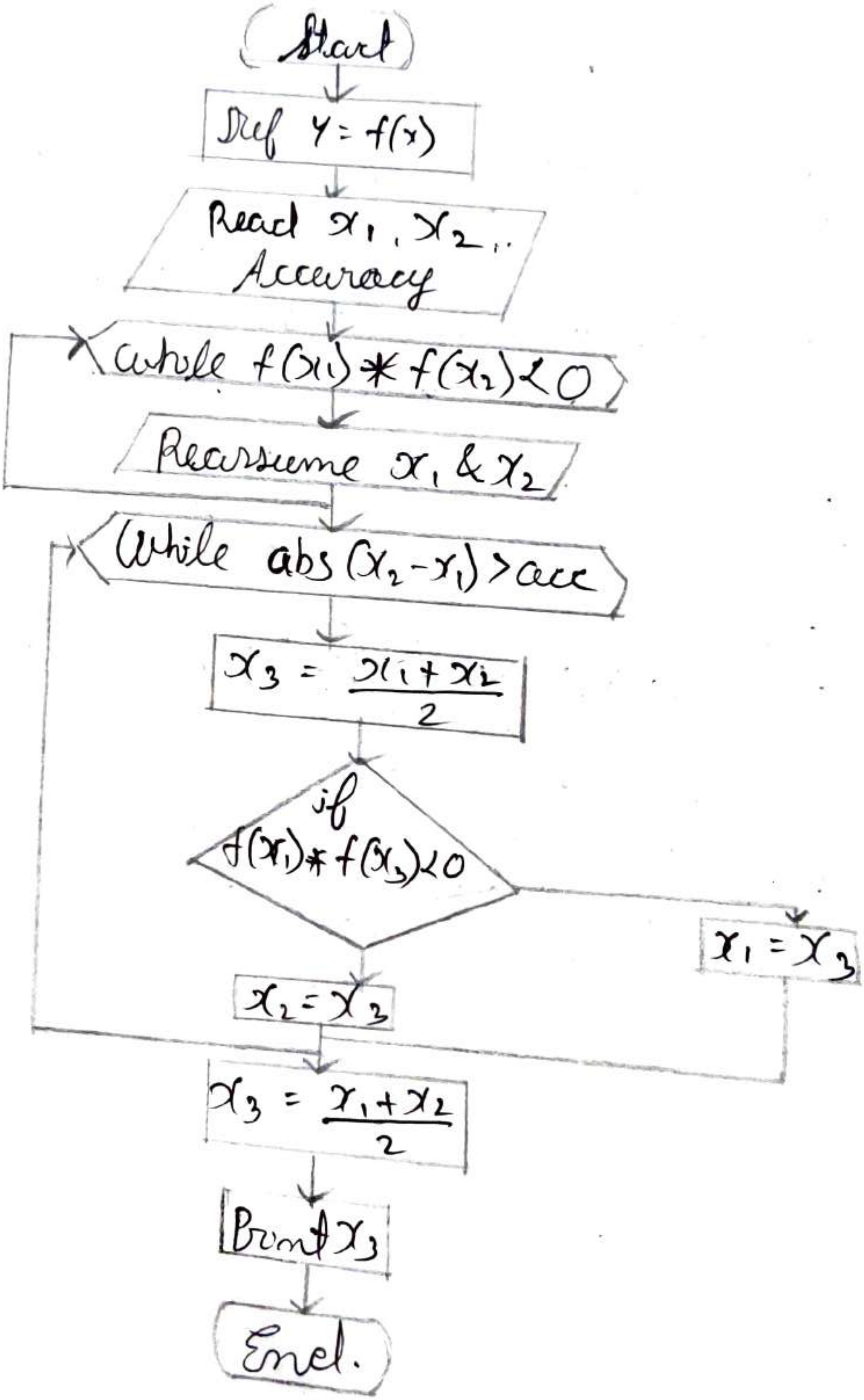
Iteration	x_1	x_2	$f(x_1)$	$f(x_2)$	x_3	$(x_2 - x_1)$
1	2	3	-9	6	2.5	1
2	2.5	3	-3.375	6	2.75	0.5
3	2.5	2.75	-3.375	0.7968	2.625	0.25
4	2.625	2.75	-1.4121	0.7968	2.687	0.125
5	2.687	2.7185	-0.3341	0.7968	2.7187	0.125
6	2.687	2.718	-0.3341	0.2209	2.7031	0.031
7	2.703	2.7189	-0.0610	0.2209	2.7109	0.015
8	2.7031	2.7109	-0.061	0.0799	2.7070	0.007
9	2.7031	2.7070	-0.061	0.009	2.7050	0.003
10	2.7056	2.7070	-0.0260	0.009	2.7060	0.002
11	2.7060	2.70703	-0.0084	0.009	2.7065	0.001

∴ Root of equation is 2.7065

Algorithm [Bisection Method]

1) Define $f(x)$ & number of iteration required 'n'.
2) Guess the two initials x_1 & x_2 such that
$$f(x_1) * f(x_2) < 0$$
also $f(x_1)$ & $f(x_2)$ must must have opposite sign.
3) Find next approximation value of root using formula $x_3 = \frac{x_1 + x_2}{2}$
4) Check $(x_2 - x_1) <$ accuracy.
5) Check $f(x_1) * f(x_3) > 0$
if yes $x_2 - x_3$
if not $x_1 - x_3$
6) Repeat step 3, 4, 5 for n times.
7) Write root of given equation x_3
8) End.

Flowchart Bisection



```
%NAME: Sarthak Abhay Patil
%ROLL NO:TYMEB228
%BATCH:B4

f=inline('(x^2+8*x-4)');
x1=input('enter the value of x1:');
x2=input('enter the value of x2:');
n=input('enter the number of iteration:');
y1=f(x1)
y2=f(x2)
while((y1*y2)>0)
    x1=input('enter the value of x1 again:');
    x2=input('enter the value of x2 again:');
    y1=f(x1)
    y2=f(x2)
end
for i=1:n
    x3=(x1+x2)/2
    y3=f(x3)
    if((y1*y3)<0)
        x2=x3
        y2=y3
    else
        x1=x3
        y1=y3
    end
end
fprintf('print root x3=%f',x3);
```

```
>> BisectionmethodAssignment2
enter the value of x1:0
enter the value of x2:1
enter the number of iteration:8

y1 =

    -4


y2 =

     5


x3 =

    0.5000


y3 =

    0.2500


x2 =

    0.5000


y2 =

    0.2500


x3 =

    0.2500


y3 =

   -1.9375


x1 =

    0.2500


[illegible]

   [illegible]
```

B) Newton - Raphson Method :-

Q) Solve $e^x \cos(x) - 1.2 = 0$ using NR Method take 4 decimal accuracy (0.0001)

→ Sol :-

1) $y = f(x) = e^x \cos(x) - 1.2$

2) Assume x_1,

Let $x_1 = 1$

$F(x_1) = 0.268$

$F'(x_1) = e^x \cos x + e^x (-\sin x)$

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

Therefore, $f'(x_1) = -0.8186$

$F''(x_1) = -2e^x \sin x_1$

$F''(x_1) = -4.5747$

3) Check by convergence criteria

$$\frac{F(x_1) * F''(x_1)}{[F'(x_1)]^2} < 2$$

$$\frac{0.268 \times (-4.574)}{(-0.8186)^2} = -1.8293$$

$\therefore -1.8293 \leq 1$

— satisfied

$\therefore x_1 = 1$

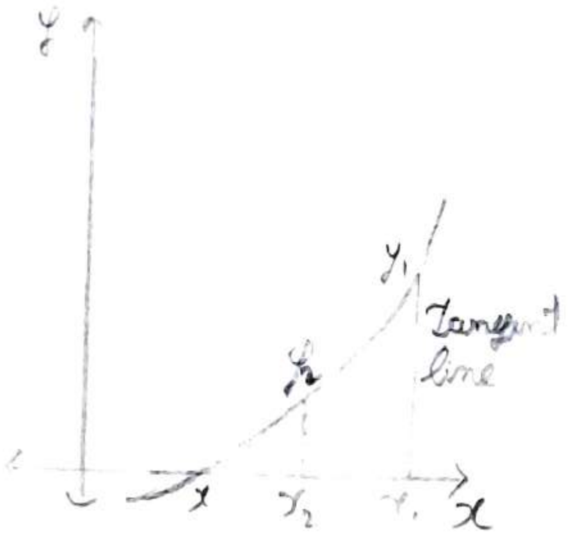
4) $x_2 = x_1 - F(x_1)/F'(x_1)$

$= 1 - (0.268/(-0.8186))$

$\therefore x_2 = 1.327$

PCCOE

Graphical Representation of Newton Raphson method :-



5) $|x_2 - x_1|$ & accuracy

Here

$|1.327 - 1| = 1.327 > 0.001$ — acc

$\therefore x_1 = x_2$

New $x_1 = 1.327$ & same procedure is repeated ~~until~~ until accuracy is achieved.

Sr No	x_1	$f(x_1)$	$f'(x_1)$	x_2	$f(x_2)$	$\lvert x_2 - x_1 \rvert$
1)	1	0.2687	-0.8187	1.3282	-0.2933	0.3282
2)	1.3282	-0.2933	-2.0284	1.2218	-0.0396	0.1064
3)	1.2218	-0.0396	-2.7571	1.2023	-1.29×10^{-3}	0.0195
4)	1.2023	-1.29×10^{-3}	-1.9057	1.2016	3.59×10^{-3}	7×10^{-4}
5)	1.2016	3.59×10^{-5}	-1.9013	1.2016	3.59×10^{-3}	0

$\therefore$ Therefore,

Root of given eqn $e^x\cos(x) - 1.2 = 0$ is 1.2016

Algorithm [~~Bisection method~~] [Newton Raphson].

1) Define $f(x)$ & number of iteration required 'n'.
2) Assume initial guess x_1
3) Calculate single & double derivatives of functions $f(x)$ i.e $f'(x)$ & $f''(x)$
4) Check convergence criteria

$$\frac{f(x_1) * f''(x_1)}{[f'(x_1)]^2} < 1 \quad \text{True}$$

if < 1 initial guess is correct else repeat step 2, 3 & 4 for correct initial guess

5) Calculate the root (approx) by using formula $x(n+1) = xn - \frac{f(xn)}{f'(xn)}$
6) if $(xn+1 - xn) <$ accuracy, then root of given equation else repeat step no 5
7) End.

Algorithm [~~Bisection method~~] [Newton Raphson]

1) Define $f(x)$ & ~~no~~ number of iteration required 'n'.

2) Assume initial guess x_1

3) Calculate single & double derivatives of functions $f(x)$ i.e $f'(x)$ & $f''(x)$

4) Check convergence criteria

$$\frac{f(x_1) * f''(x_1)}{[f'(x_1)]^2} \le 1 \quad \text{True}$$

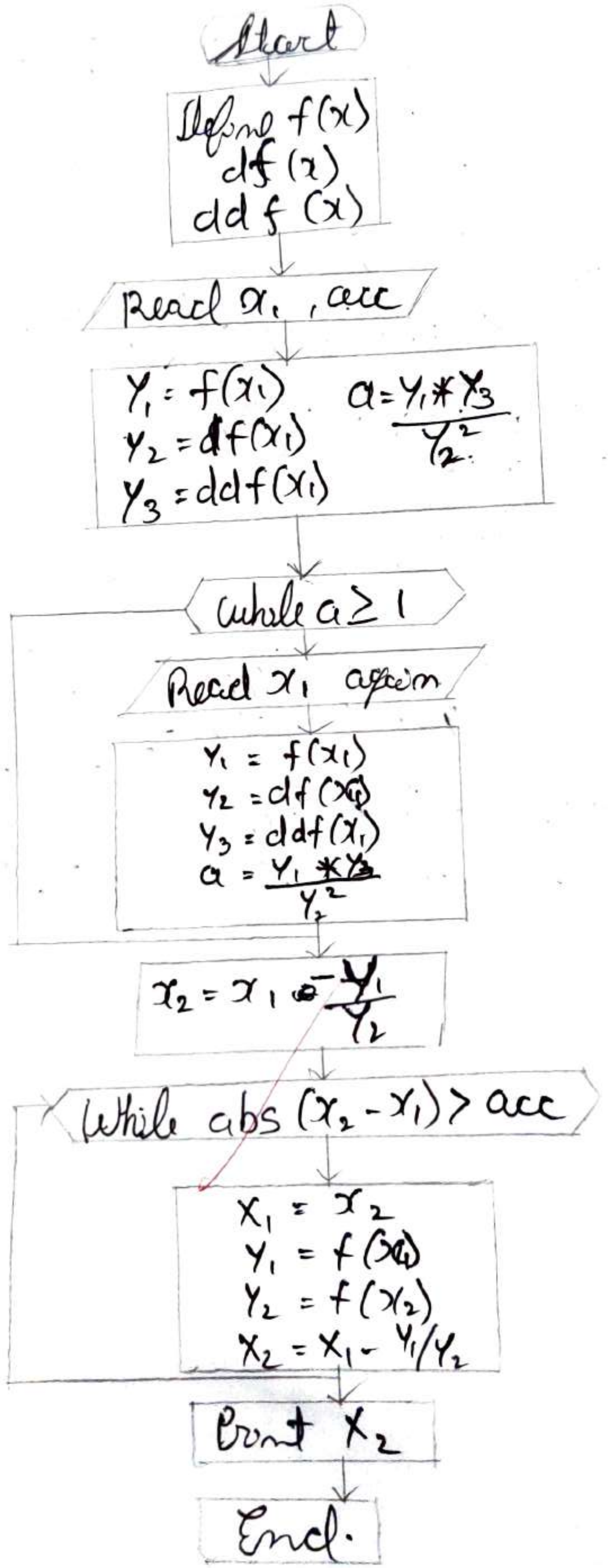
4) if <1 initial guess is correct else repeat step 2, 3 & 4 for correct initial guess.

5) Calculate the root (approx) by using formula $x(n+1) = xn - \frac{f(xn)}{f'(xn)}$

6) if $(xn+1 - x(n)) <$ Accuracy, then root of given equation else repeat step no 5

7) End.

Flowchart. Newton Raphson



```
%NAME: Sarthak Abhay Patil
%ROLL NO:TYMEB228
%BATCH:B4

f=inline('exp(x)*cos(x)-1.2');
df=inline('exp(x)*(cos(x)-sin(x))');
ddf=inline('(-2)*exp(x)*sin(x)');

x1=input('Enter initial value:');
acc=input('Enter the accuracy:');
while f(x1)*ddf(x1)/df(x1)^2>1
    x1=input('Enter initial value again');
end
x2=x1-f(x1)/df(x1)
while abs(x2-x1)>acc
    x1=x2
    x2=x1-f(x1)/df(x1)
end
fprintf('Root of equation x2= %f',x2)
```

```
>> NewtonsRaphsonAssignment2
Enter initial value:1
Enter the accuracy:0.0001

x2 =

    1.3282


x1 =

    1.3282


x2 =

    1.2[illegible]18


x1 =

    [illegible]


x2 =

    [illegible]


x1 =

    [illegible]


[illegible]

    [illegible]


[illegible]

    [illegible]


[illegible]

    [illegible]

[illegible] 1.2[illegible]
```

Assignment No 3

Simultaneous Equation:-
In Numerical methods & optimization the simultaneous equation method refers to solving a system of equations simultaneously to find the values of unknown variables that satisfys all equations in system.

a) Gauss - Elimination Method :-
- This Method is used to solve system of linear equations by systematically transforming equations to matrix representation by pivoting help & then back substituting to find the solution. It is versatile but can be computationally intensive for large systems.

b) Thomas Algorithm for Tri-diagonal Matrix:-
- Tri diagonal method is specifically designed for solving systems of equations where the coefficient matrix is tridiagonal.
- Non zero element only on main & adjacent to main diagonal.

c) Gauss - Seidel Method:-
It is an iterative technique for solving systems of simultaneous linear ~~eq~~ equations
It is useful for large & sparse systems.

Numericals :-

i) Gauss Elimination Method

$2x + 4y + 3z = 13$
$3x + 6y + z = 16$
$1x + 3y + 2z = 9$

Re-arranging the above equation

$3x + 6y + z = 16$ —— ①
$2x + 4y + 3z = 13$ —— ②
$1x + 3y + 2z = 9$ —— ③

Matrix

$$\begin{bmatrix} 3 & 6 & 1 \\ 2 & 4 & 3 \\ 1 & 3 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 16 \\ 13 \\ 9 \end{bmatrix}$$

$R_2 \rightarrow R_2 - \frac{R_1}{a_{11}} \times a_{21}$ $\qquad$ $R_3 \rightarrow R_3 - \frac{R_1}{a_{11}} \times a_{31}$

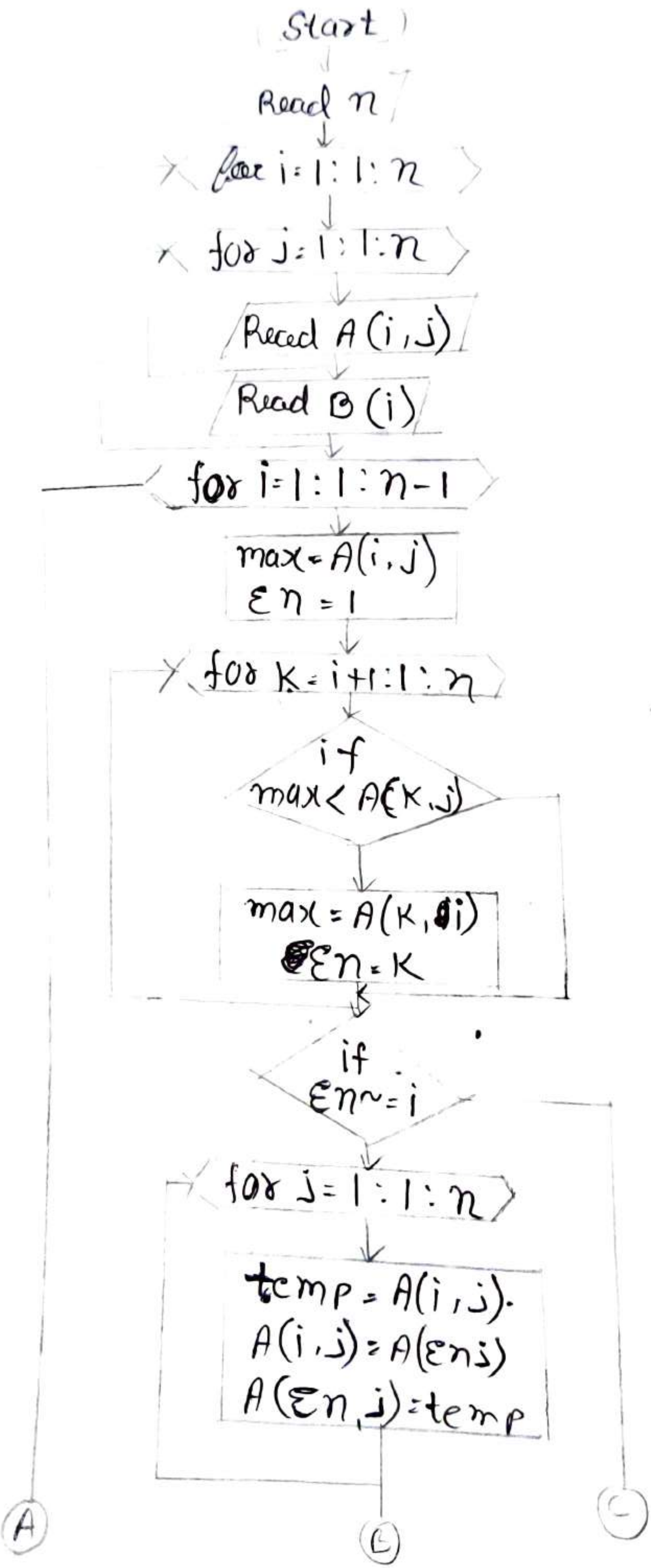
1st pivoting

$$\begin{bmatrix} 3 & 6 & 1 \\ 2 & 4 & 3 \\ 1 & 3 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 16 \\ 3 \\ 9 \end{bmatrix}$$

$R_3 \rightarrow R_3' - \frac{R_2}{a_{22}} \times a_{32}$

~~2nd Pivoting~~

1) Flowchart for Gauss Elimination Method



2^{nd} pivoting

$$\begin{bmatrix} 3 & 6 & 1 \\ 0 & 0 & 7/3 \\ 0 & 1 & 5/3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 16 \\ 7/3 \\ 11/3 \end{bmatrix}$$

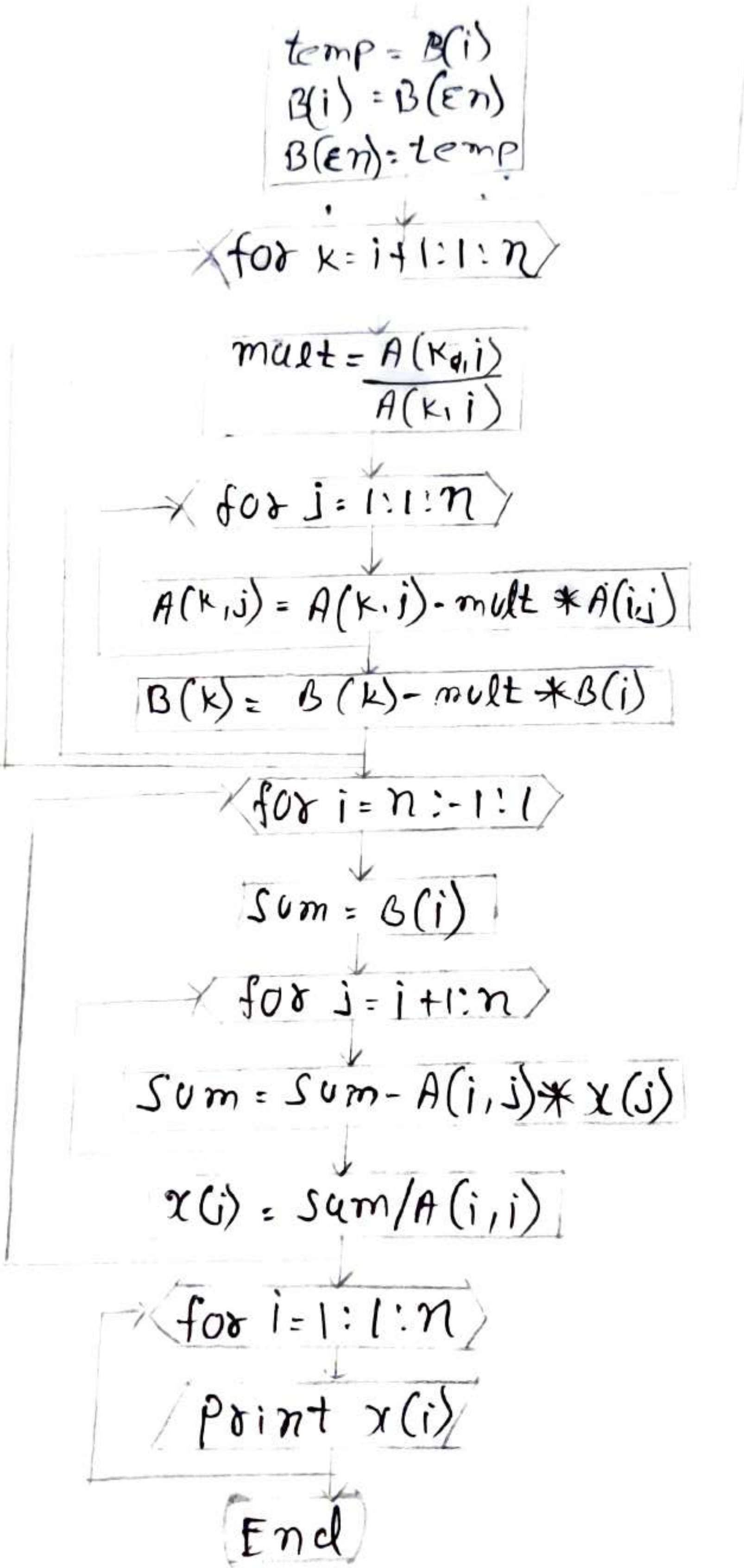
Equations

$$3x + 6y + z = 16$$
$$y + 5/3\, z = 11/3$$
$$7/3\, z = 7/3$$

$\therefore x = 1$

$y = 2$

$z = 1$


temp = B(i)
B(i) = B(En)
B(En) = temp
for k = i+1:1:n
mult = A(k,i)/A(k,i)
for j = 1:1:n
A(k,j) = A(k,j) - mult * A(i,j)
B(k) = B(k) - mult * B(i)
for i = n:-1:1
sum = B(i)
for j = i+1:n
sum = sum - A(i,j) * x(j)
x(i) = sum/A(i,i)
for i = 1:1:n
print x(i)
End

```
% Name : Sarthak Abhay PATIL
% Roll no. : TYMEB228
% Div. : B / B4

%INPUT
n = input('Enter the no. of the Equation= ');
for i=1:1:n
    for j=1:1:n
        a(i,j)= input('Enter the value of a(i,j)= ');
    end
    b(i)= input('Enter the value of b(i)= ');
end
for i=1:1:n-1
    max= abs(a(i,i));
    Rn=i;

    for k=i+1:1:n
        if max<abs(a(k,i))
            max= abs(a(k,i));
            Rn= k;
        end
    end
    if Rn~=i
        for j=1:1:n
             temp  = a(i,j);
            a(i,j) = a(Rn,j);
            a(Rn,j)= temp;
        end
        temp = b(i);
        b(i) = b(Rn);
        b(Rn)= temp;
    end
end
Acc = input('Enter the value of Accuracy= ');
err = 1;
xold = zeros(n,1);
xnew = zeros(n,1);

while err>Acc
    for i=1:1:n
        xold(i)=xnew(i);
    end
    for i=1:1:n
        term = b(i);
        for j= 1:1:n
            if j~=i
                if j>i
                    term = term-a(i,j)*xold(j);
                else
                    term = term-a(i,j)*xnew(j);
                end
```

```
            end
        end
        xnew(i[illegible] = (term/a[illegible]
        e(i[illegible] abs(xnew[illegible]x[illegible]d
    end

    err = e(1[illegible]
    [illegible]
        [illegible]
            [illegible]
        end
    end
end
[illegible]
    [illegible]
end
```

[illegible]

2) Tri - diagonal matrix Algorithm method:-
- Solve the foll tri - diagonal system & find temp T_1, T_2, T_3, T_4 in °C

$$\begin{bmatrix} 2.04 & -1 & 0 & 0 \\ -1 & 2.04 & -1 & 0 \\ 0 & -1 & 2.04 & -1 \\ 0 & 0 & -1 & 2.04 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} 40.8 \\ 0.8 \\ 0.8 \\ 200.8 \end{bmatrix}$$

→ $a = \begin{bmatrix} 0 \\ -1 \\ -1 \\ -1 \end{bmatrix}$ $b = \begin{bmatrix} 2.04 \\ 2.04 \\ 2.04 \\ 2.04 \end{bmatrix}$ $c = \begin{bmatrix} -1 \\ -1 \\ -1 \\ 0 \end{bmatrix}$ $d = \begin{bmatrix} 40.8 \\ 0.8 \\ 0.8 \\ 200.8 \end{bmatrix}$

$i : 2 : 1 : n$

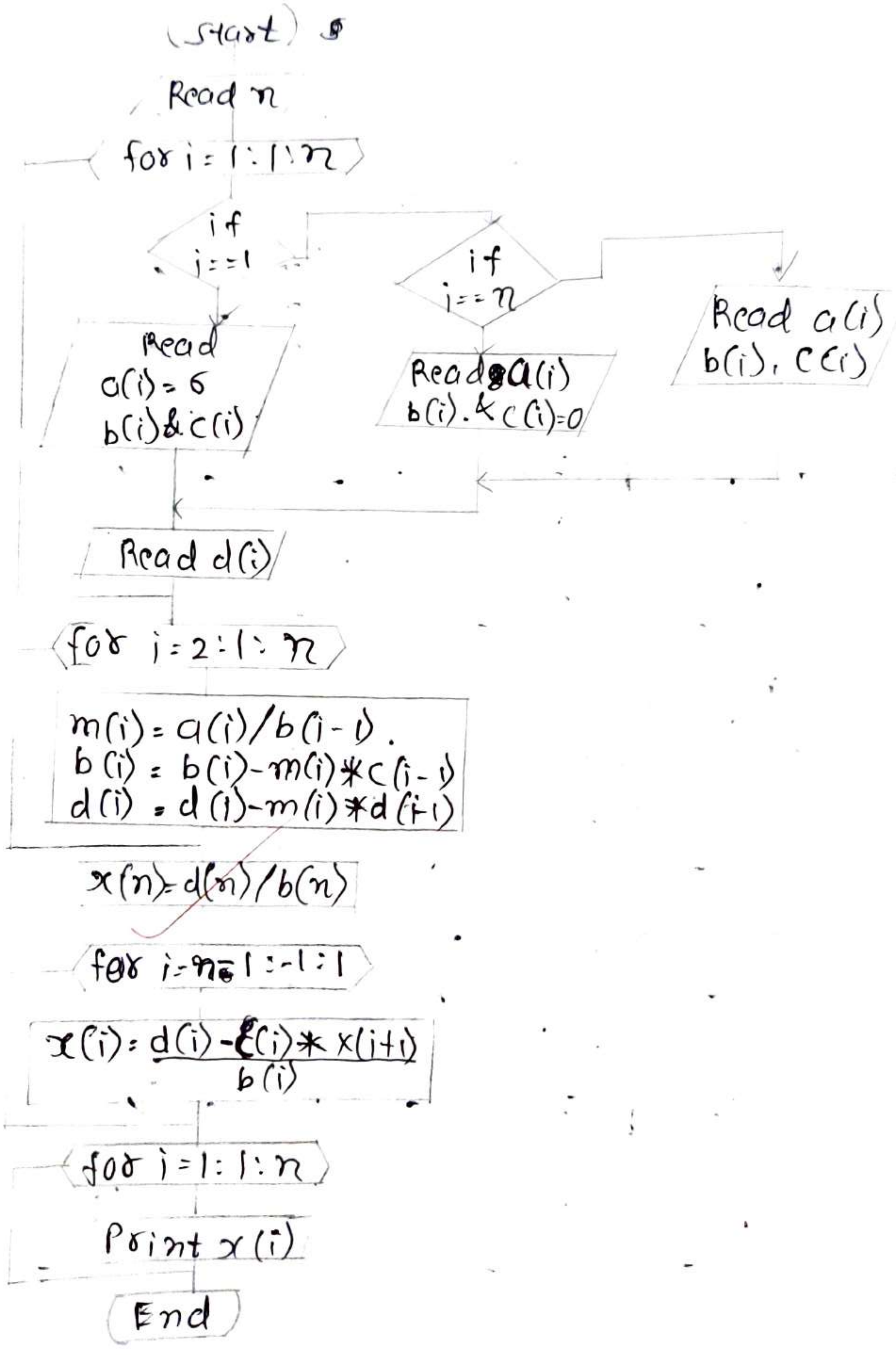
$m(i) = \frac{a(i)}{b(i-1)}$ $m_2 = \frac{-1}{2.04}$

$b(i) = b(i) - m(i) \times c(i-1)$
$= -2.04 - \left(\frac{-1}{2.04}\right) \times (-1) = 1.549$

$d(i) = d(i) - m(i) \times d(i-1)$
$= 0.8 - \left(\frac{-1}{2.04}\right) \times 40.8 = 20.8$

$a = \begin{bmatrix} 0 \\ 0 \\ -1 \\ -1 \end{bmatrix}$ $b = \begin{bmatrix} 2.04 \\ 1.549 \\ 2.04 \\ 2.04 \end{bmatrix}$ $c = \begin{bmatrix} -1 \\ -1 \\ -1 \\ 0 \end{bmatrix}$ $d = \begin{bmatrix} 40.8 \\ 20.8 \\ 0.8 \\ 200.8 \end{bmatrix}$

2) Flowchart for TDMA



```
n=input('enter no of equations n=');
for i=1:1:n
    if i==1
        a(i)=0;
        b(i)=input('enter B matrix element=');
        c(i)=input('enter C matrix element='
    else
        if i==n
            a(i)=input('enter A matrix element=');
            b(i)=input('enter B matrix element=');
            c(i)=0;
        else
            a(i)=input('enter A matrix element=');
            b(i)=input('enter B matrix element=');
            c(i)=input('enter C matrix element=');
        end
    end
    d(i)=input('enter D matrix element=');
end
for i=2:1:n
    m(i)=a(i)/b(i-1);
    b(i)=b(i)-m(i)*c(i-1);
    d(i)=d(i)-m(i)*d(i-1);
end
x(n)=d(n)/b(n)
for i=n-1:-1:1
    x(i)=(d(i)-c(i)*x(i+1))/b(i);
end
for i=1:1:n
    fprintf('\nX(%d)=%f',i,x(i));
end

% OUTPUT
% TDMA
% enter no of equations n=4
% enter B matrix element=2.04
% enter C matrix element=-1
% enter D matrix element=40.8
% enter A matrix element=-1
% enter B matrix element=2.04
% enter C matrix element=-1
% enter D matrix element=0.8
% enter A matrix element=-1
% enter B matrix element=2.04
% enter C matrix element=-1
% enter D matrix element=0.8
% enter A matrix element=-1
```

```
% enter B matrix element=2.04
% enter D matrix element=200.8
%
% x =
%
%    31.6887   23.8450   16.1550  159.4795
%
%
% X(1)=65.969834
X(2)=93.776462
X(3)=124.538228
X(4)=159.479524>>
```

$i = 3$, $\cancel{m=3}$ $m_3 = \frac{a(3)}{b(3-1)} = \frac{-1}{1.544}$

$b(3) = 2.04 - \left(\frac{-1}{1.549}\right) \times (-1) = 1.394$

$d(3) = 0.8 - \left(\frac{-1}{1.544}\right) \times 20.8 = 14.228$

$a = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix}$ $b = \begin{bmatrix} 2.04 \\ 1.549 \\ 1.394 \\ 2.04 \end{bmatrix}$ $c = \begin{bmatrix} -1 \\ -1 \\ -1 \\ 0 \end{bmatrix}$ $d = \begin{bmatrix} 40.8 \\ 20.8 \\ 14.228 \\ 200.8 \end{bmatrix}$

$i = 4$ $m_4 = \frac{a(4)}{b(4-1)} = \frac{-1}{1.394}$

$b(4) = 2.04 - \left(\frac{-1}{1.394}\right) \times (-1) = 1.322$

$d(4) = 200.8 - \left(\frac{-1}{1.394}\right) \times 14.228 = 211.006$

$a = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ $b = \begin{bmatrix} 2.04 \\ 1.544 \\ 1.394 \\ 1.322 \end{bmatrix}$ $c = \begin{bmatrix} -1 \\ -1 \\ -1 \\ 0 \end{bmatrix}$ $d = \begin{bmatrix} 40.8 \\ 20.8 \\ 14.228 \\ 211.006 \end{bmatrix}$

$$x(n) = \frac{d(n)}{b(n)}$$

$$T_4 = x(4) = \frac{d(4)}{b(4)} = \frac{211.006}{1.322}$$

$$T_4 = 159.611\,°C$$

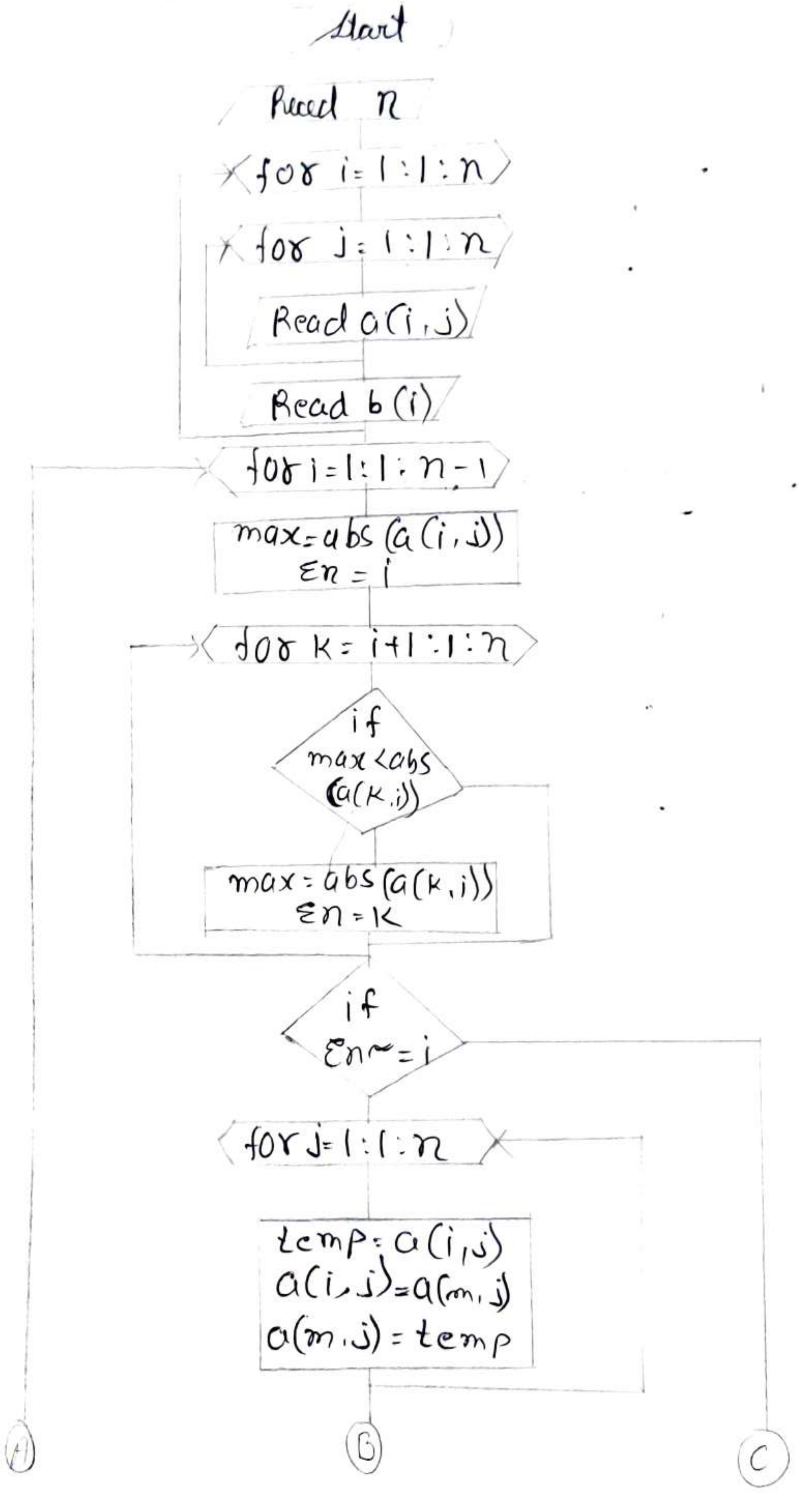
$$T(i) = \frac{d(i) - c(i) \times T(i+1)}{b(i)}$$

$$T(3) = 124.70\,°C$$

$$T(2) = 93.77\,°C$$

$$T(1) = 63.96\,°C$$

3) Flowchart for Gauss Seidal Method



3) Gauss Seidel method :-

- Solve the foll upto 0.001 accuracy

$x_1 + 20x_2 + 9x_3 = -23$

$2x_1 - 7x_2 - 20x_3 = -57$

$20x_1 + 2x_2 + 6x_3 = 28$

→ Pivoting

$20x_1 + 2x_2 + 6x_3 = 28$ —— ①

$x_1 + 20x_2 + 9x_3 = -23$ —— ②

$2x_1 - 7x_2 - 20x_3 = -51$ —— ③

- from eqn ① ② ③

$$x_1 = \frac{1}{20}(28 - 2x_2 - 6x_3)$$

$$x_2 = \frac{1}{20}(-23 - x_1 - 9x_3)$$

$$x_3 = \frac{1}{-20}(-57 - 2x + 7x_2)$$

Iteration ①

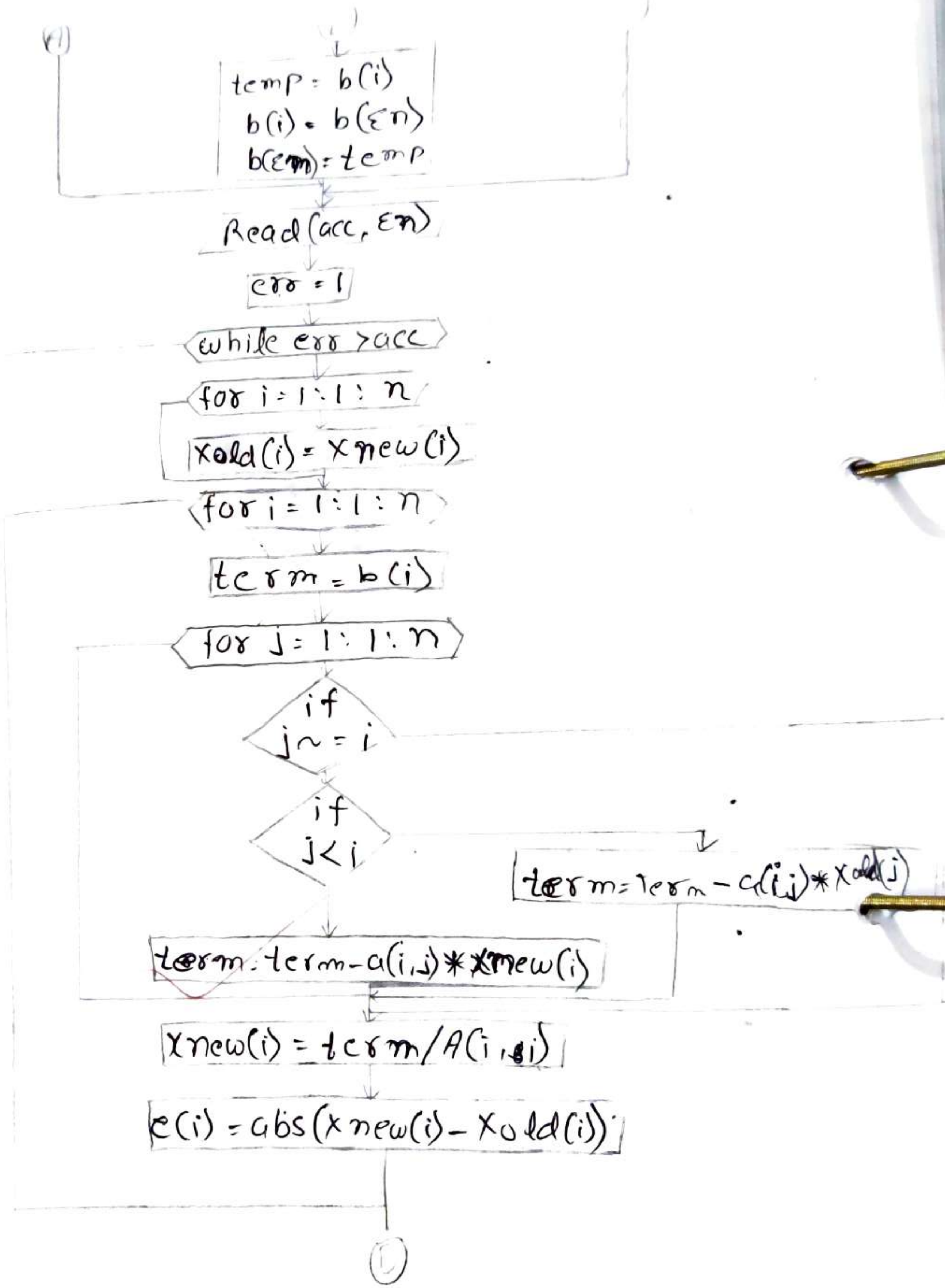
assume $x_1 = 0,\ x_2 = 0,\ x_3 = 0$

$$\therefore x_1 = \frac{1}{20}(28 - 2(0) - 6(0)) = 1.4$$

$x_2 = -1.22$

$x_3 = 3.417$

PCCOE



temp = b(i)
b(i) = b(εn)
b(εm) = temp
Read(acc, εn)
err = 1
while err > acc
for i = 1:1:n
xold(i) = xnew(i)
for i = 1:1:n
term = b(i)
for j = 1:1:n
if j~=i
if j<i
term = term - a(i,j) * xold(j)
term = term - a(i,j) * xnew(i)
xnew(i) = term/A(i,i)
e(i) = abs(xnew(i) - xold(i))

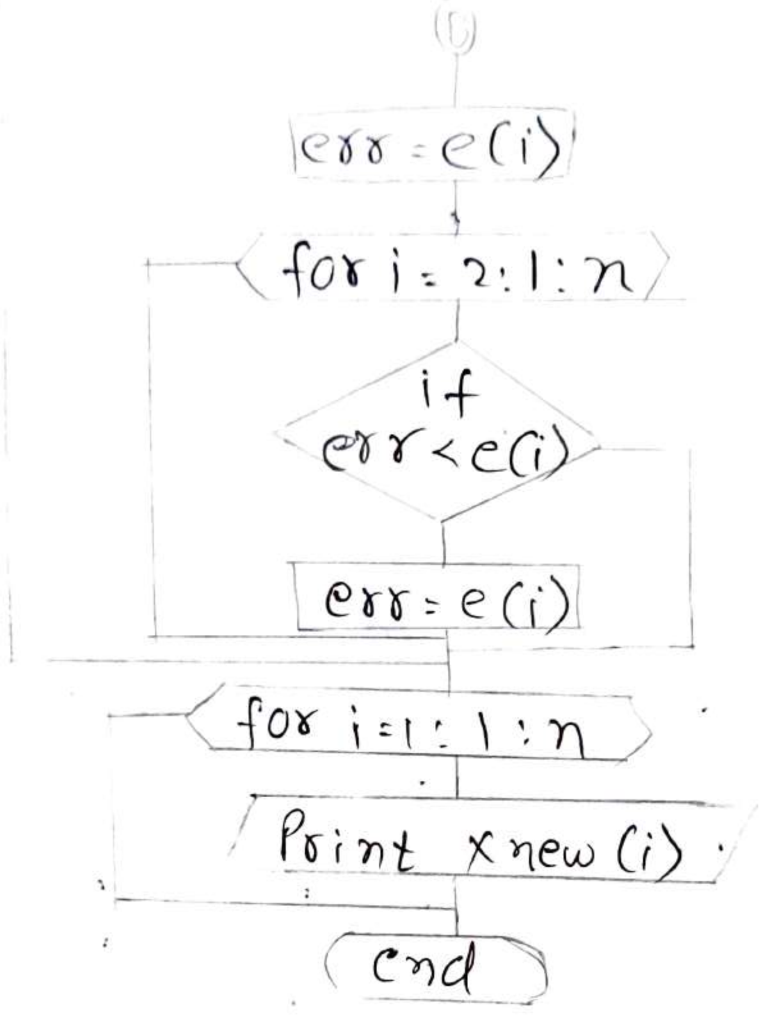
$|x_1 new - x old| = 1.4 > 0.001$
$|x_2 new - x old| = 1.22 > 0.001$
$|x_3 new - x old| = 3.417 > 0.001$

— doesn't match with accuracy

then go for next iteration

itr no	x_1	x_2	x_3
1)	0	0	0
2)	1.4	-1.22	3.417
3)	0.446	-2.7125	3.849
4)	0.6165	-2.9079	3.919
5)	0.5149	-2.9394	3.930
6)	0.5148	-2.9443	3.9320
7)	0.51483	-2.9451	3.9322

PCCOE



err = e(i)
for i = 2:1:n
if err < e(i)
err = e(i)
for i = 1:1:n
Print xnew(i)
end

```
n=input('enter number or equation');
for i=1:1:n
    for j=1:1:n
        a(i,j)=input('Enter matrix A elements=');
    end
        b(i)=input('Enter matrix B elements=');
end
  for i=1:1:n-1
max = abs(a(i,i));
rn = i;
for k = i+1:1:n
    if max<abs(a(k,i));
        max = abs(a(k,i));
        rn=k;
    end
    if rn == i;
        for j=1:1:n
            temp=a(i,j);
            a(i,j)=a(rn,j);
            a(rn,j)=temp;
        end;
        temp = b(i);
        b(i)=b(rn);
        b(rn)=temp;
    end
    end
end
acc = input('Enter accuracy = ');
err=1
xold=zeros(n,1);
xnew=zeros(n,1);
while err>acc
    for i=1:1:n
        xold(i)=xnew(i);
    end
    for i=1:1:n
        term=b(i);
        for j = 1:1:n
    if j~=i
        if j<1
            term=term-a(i,j)*xold(j);
        else
            term=term-a(i,j)*xnew(j);
        end
    end
        end
        xnew(i)=((term)/a(i,j));
        e(i)=abs(xold(i)-xnew(i));
        fprintf('\tx(%d)=%f',i,xnew(i));
    end
    err= e(1)
```

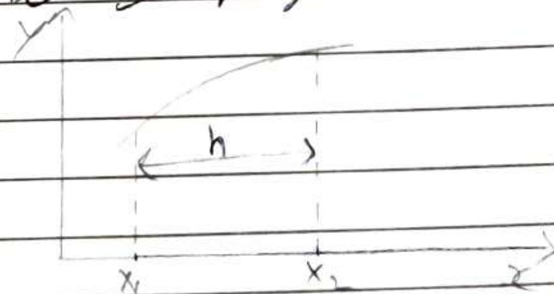
```
    for i=2:1:n
        if err<e(i)
            err=e(i);
        end
    end
    fprintf('\n');
end
enter number or equation3
Enter matrix A elements=1
Enter matrix A elements=2
Enter matrix A elements=20
Enter matrix B elements=20
Enter matrix A elements=-7
Enter matrix A elements=2
Enter matrix A elements=9
Enter matrix B elements=-20
Enter matrix A elements=6
Enter matrix A elements=-23
Enter matrix A elements=-57
Enter matrix B elements=28
Enter accuracy = 0.001
err =
 1
 x(1)=1.000000 x(2)=-1.444444 x(3)=0.196881
err =
 1
 x(1)=0.947563 x(2)=-1.682110 x(3)=0.287261
err =
 0.0524
 x(1)=0.880950 x(2)=-1.824301 x(3)=0.337625
err =
 0.0666
 x(1)=0.844805 x(2)=-1.902776 x(3)=0.365486
err =
 0.0361
 x(1)=0.824792 x(2)=-1.946203 x(3)=0.380902
err =
 0.0200
 x(1)=0.81[illegible]18 x(2)=-1.970232 x(3)=0.389433
err =
 0.0111
 x(1)=0.807[illegible]91 x(2)=-1.[illegible]83529 x(3)=0.394153
err =
 0.006[illegible]
 x(1)=0.804200 x(2)=-1.990886 x(3)=0.396765
err =
 0.0034
 x(1)=0.802[illegible]24 x(2)=-1.99495[illegible] x(3)=0.398210
err =
 0.0019
 x(1)=0.801[illegible] x(2)=-1.[illegible] x(3)=0.399009
```

Assignment no 04

Numerical Integration:-

- Numerical integration is the method to estimate the area under a curve when it is hard to find the exact answer.
- It breaks the curve into small parts & approximate the area of each part, then add them up for overall estimate.

1) Trapezoidal method
- it is simple but less accurate compared to other methods
- Approximates the integral by dividing the area under curve into trapezoid.

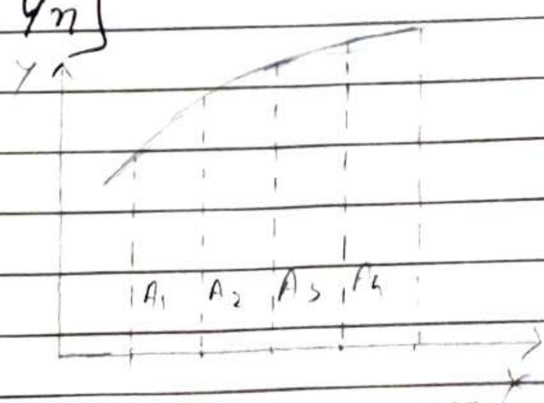


h = strip ~~cell~~ size
$A = h/2\ [Y_1 + Y_n]$
- here $Y = f(x)$

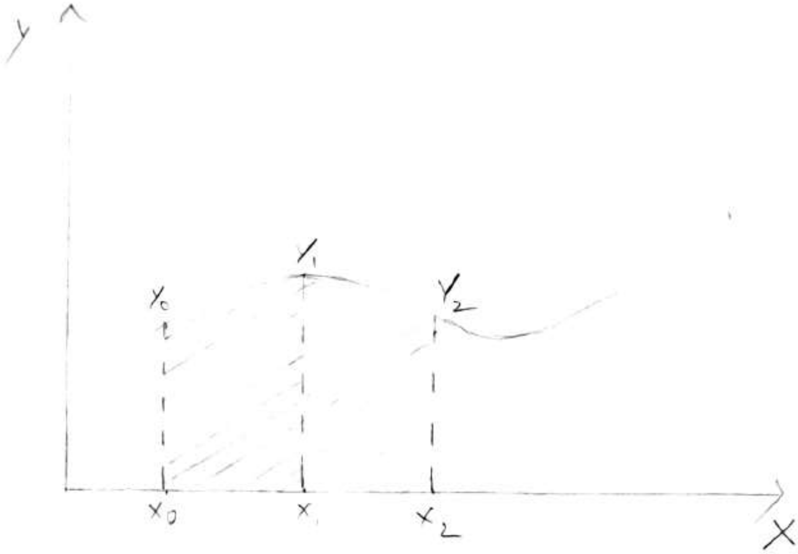
When there are multiple areas then:-

$I : A_1 + A_2 + A_3 + A_4 - \cdots A_n$

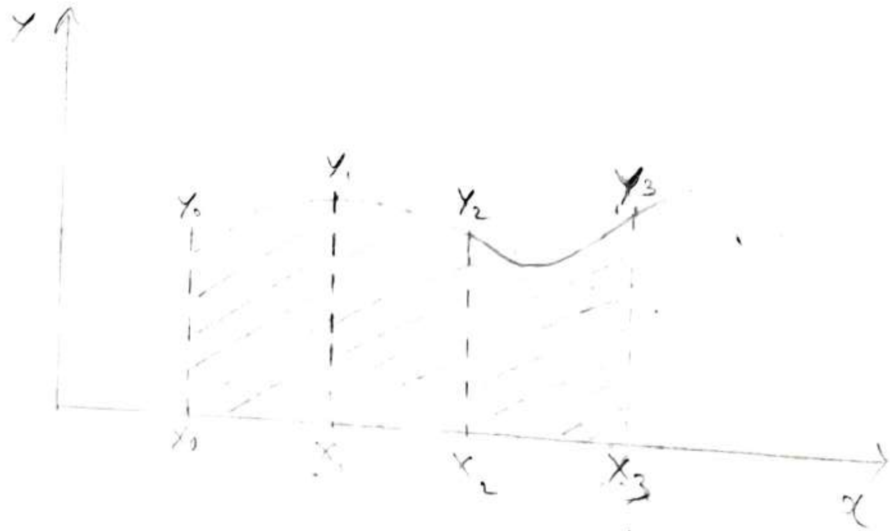
$I : \frac{h}{2} [Y_1 + 2Y_2 + 2Y_3 + 2Y_4 + Y_n]$



2) Simpsons 1/3rd rule



3) Simpsons 3/8th rule

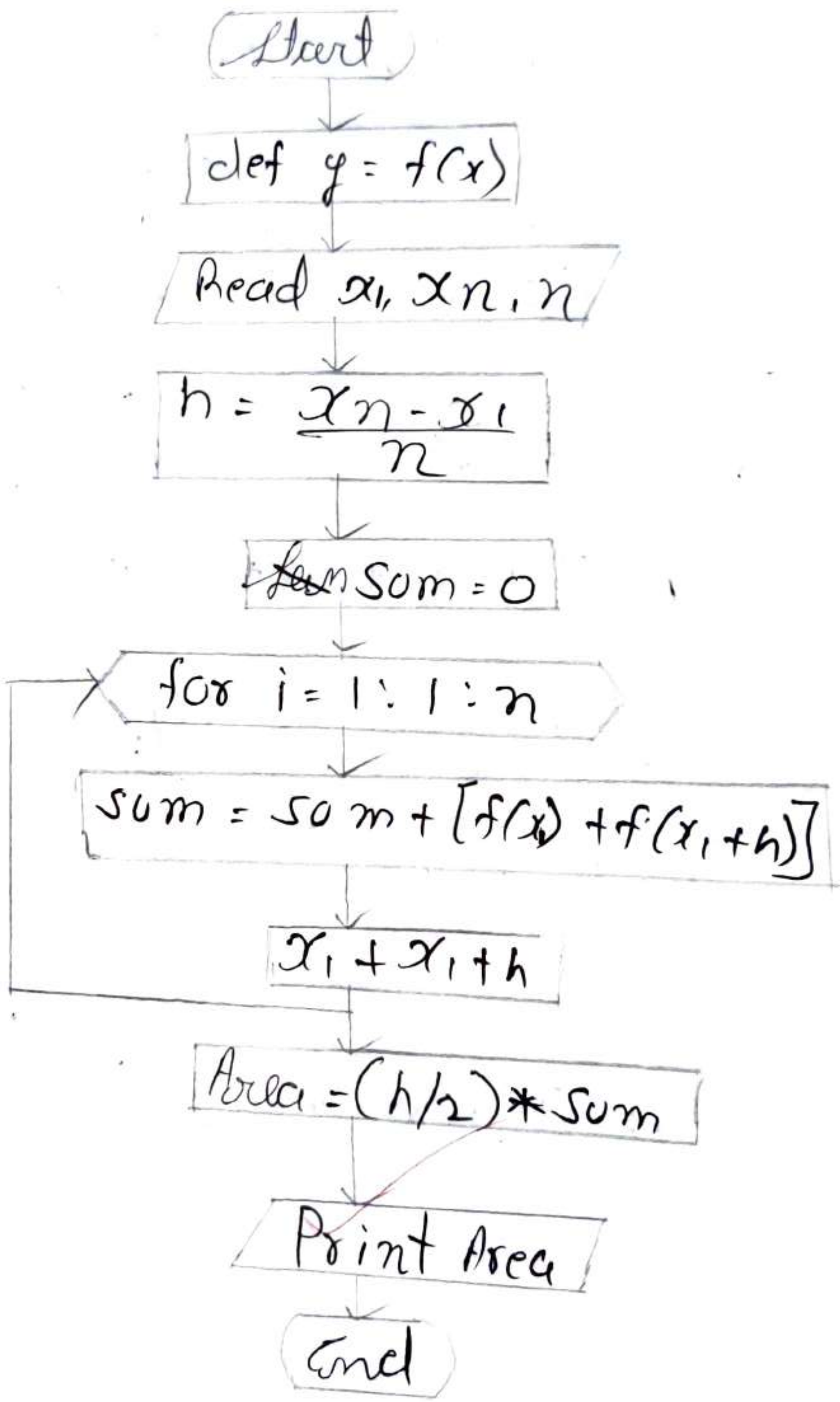


2) Simpson's 1/3th rule method :

- Simpson's 1/3rd rule uses quadratic interpolations to approximate the integral this more accurate than the trapezoidal rule & require fewer function evaluation
- A polynomial of second order so that different of order higher than second.
- here $A = \frac{h}{3}[Y_1 + 4Y_2 + 2Y_3 + 4Y_4 + Y_n]$

3) Simpson's 3/8th rule method:
- Simpsons 3/8th rule is similar to simpsons 1/3th rule but uses cubic interpolation.
- it provide even higher accuracy but require more function evaluation.
- here $A = \frac{3h}{8}[Y_0 + 3Y_1 + 3Y_2 + Y_3]$

Flowchart for Trapezoidal Rule



```
% Name : sarthak PATIL
% Roll no. : TYMEB228
% Batch : B4
f = inline('3*(0.4+0.004*T)');
x1 = input('Enter the value of x1 = ');
xn = input('Enter the value of xn = ');
n =  input('Enter the value of n = ');

h = (xn-x1)/n;
Area = 0;
for i=1:1:n
    Area = Area+(f(x1)+f(x1+h));
    x1 = (x1+h);
end
Area = (h/2)*Area;
fprintf('The value Area is =%f',Area )

%OUTPUT
Enter the value of x1 = 25
Enter the value of xn = 125
Enter the value of n = 4
The value Area is =210.000000>>
```

Q) During a certain process the specific heat capacity of system is given by C i.e $C = [0.4 + 0.004T]$. Find the heat transferred when temp changes from 25°C to 125°C. the mass of gas is 3kg. take $n = 4$ & solve by :- 1) Trapezoidal method
2) Simpson's 1/3rd rule
3) Simpson's 3/8th rule

→ Given data :- $T_1 = 25°C$
$T_n = 125°C$
$C = [0.4 + 0.004T]$ kJ/kg°C
$m = 3kg$

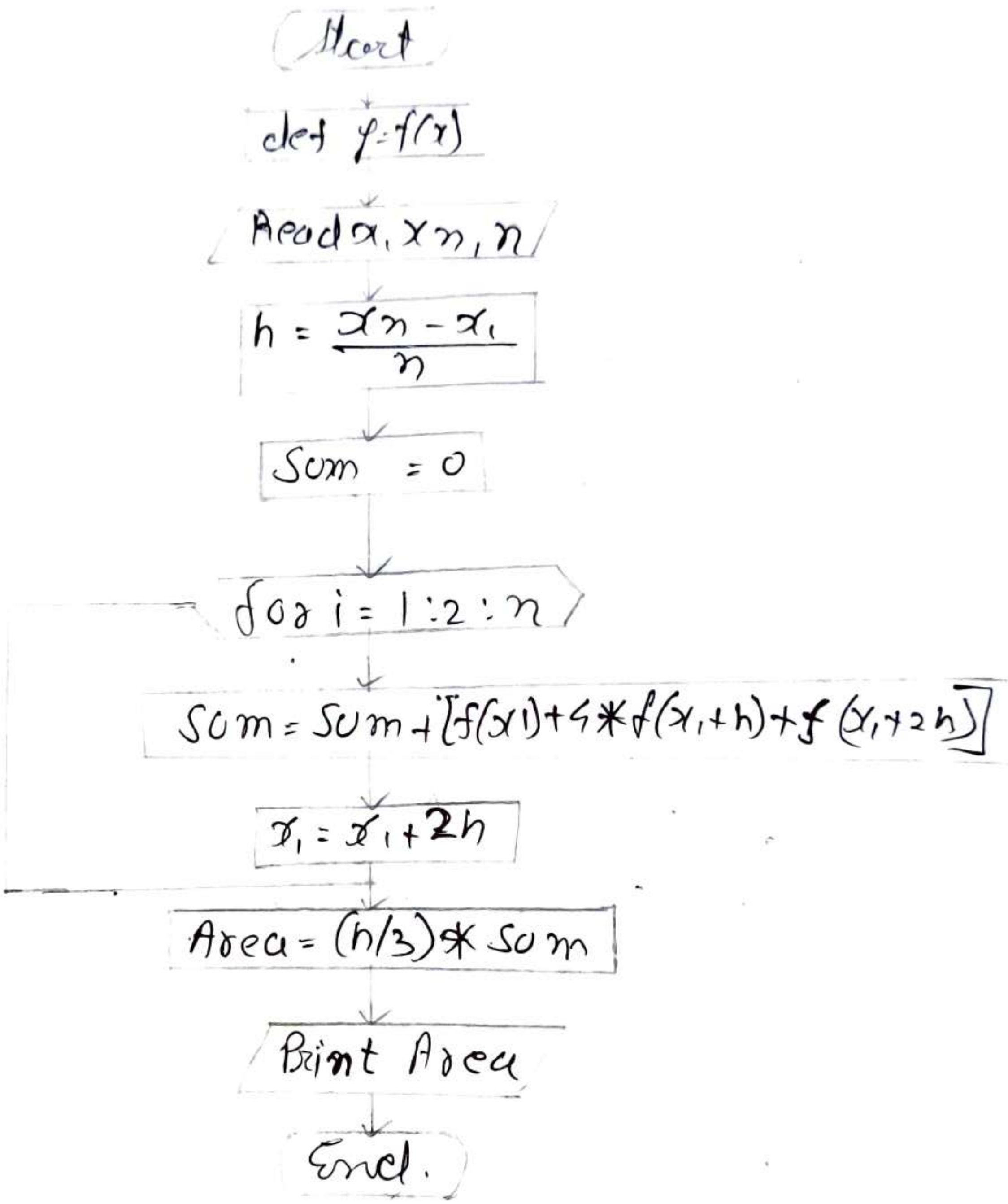
We know,
$$Q = \int_{T_1}^{T_n} m\, C_p\, \Delta T$$
$$\therefore Q = \int_{25}^{125} 3 \times [0.4 + 0.004T]\, \Delta T$$
$$y = f(T) = 3[0.4 + 0.004T]$$

$T_1 = 25°C$
$T_2 = T_1 + h = 50°C$
$T_3 = 75°C$
$T_4 = 100°C$
$T_n = 125°C$

$$h = \frac{T_n - T_1}{n} = \frac{125 - 25}{4} = 25$$

$Y_1 = f(T_1) = 3(0.4 + 0.004 \times 25) = 1.50$
$Y_2 = 1.80$
$Y_3 = 2.10$
$Y_4 = 2.40$
$Y_n = 2.70$

flowchart for simpson 1/3rd rule



```
% Name :sarthak PATIL
% Roll no. : TYMEB228
f = inline('3*(0.4+0.004*T)');
x1 = input('Enter the value of x1 = ');
xn = input('Enter the value of xn = ');
n =  input('Enter the value of n = ');

h = (xn-x1)/n;
Area = 0;
for i=1:2:n
    Area = Area+(f(x1)+ 4*f(x1+h)+ f(x1+2*h));
    x1 = (x1+2*h);
end
Area = (h/3)*Area;
fprintf('The value Area is =%f',Area )

%OUTPUT
Enter the value of x1 = 25
Enter the value of xn = 125
Enter the value of n = 4
The value Area is =210.000000
```

- By Trapezoidal rule

$$A = I = Q = \frac{h}{2}\left[Y_1 + 2Y_2 + 2Y_3 + 2Y_4 + Y_n\right]$$

$$= \frac{25}{2}\left[1.5 + (2\times1.8) + (2\times2.10) + 2(2.40) + 2.7\right]$$

$$= 210$$

- By Simpsons 1/3rd Rule

$$A = I = Q = \frac{h}{3}\left[Y_1 + 4Y_2 + 2Y_3 + 4Y_4 + Y_n\right]$$

$$= \frac{25}{3}\left[1.5 + 4(1.8) + 2(2.1) + 4(2.4) + 2.7\right]$$

$$= 210$$

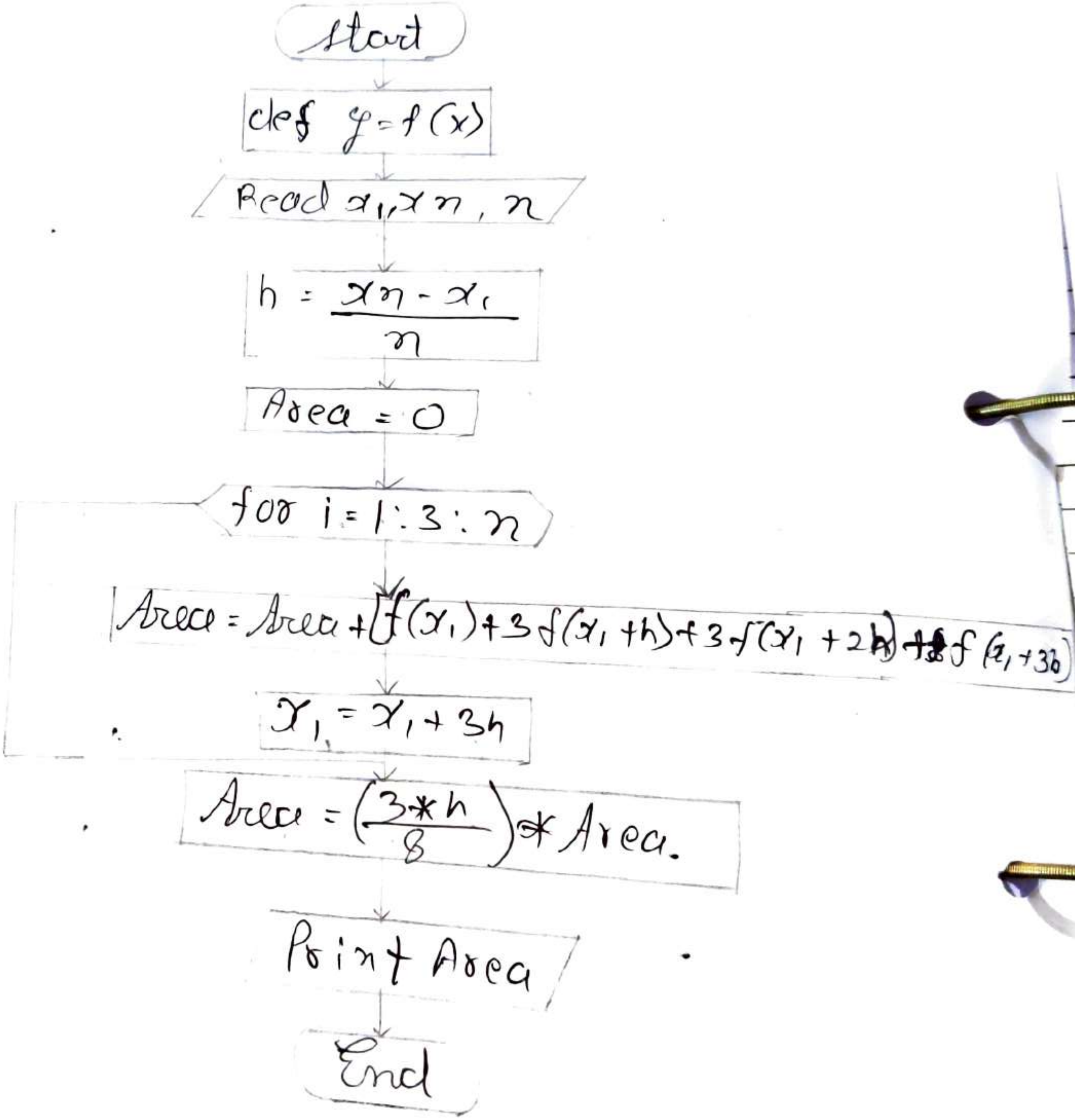
- By Simpsons 3/8th rule

$$A = I = Q = \frac{3h}{8}\left[(Y_1 + Y_n) + 3(Y_2 + Y_3) + 2(Y_4)\right]$$

$$= \frac{3\times25}{8}\left[(1.5 + 2.7) + 3(1.8 + 2.1) + 2(2.4)\right]$$

$$= 210$$

Flow chart for simpsons 3/8th rule.



```
%Name - sarthak Patil
%Roll no - TYMEB228
%Batch -B4
f = inline('3*(0.4+0.004*T)');
x1 = input('Enter the value of x1 = ');
xn = input('Enter the value of xn = ');
n =  input('Enter the value of n = ');

h = (xn-x1)/n;
Area = 0;
for i=1:3:n
    Area = Area+(f(x1)+ 3*f(x1+h)+ 3*f(x1+2*h)+ f(x1+3*h));
    x1 = (x1+3*h);
end
Area = (3*h/8)*Area;
fprintf('The value Area is =%f',Area )

simpsonsrule3by8
Enter the value of x1 = 25
Enter the value of xn = 125
Enter the value of n = 6
The value Area is =210.000000>>
```

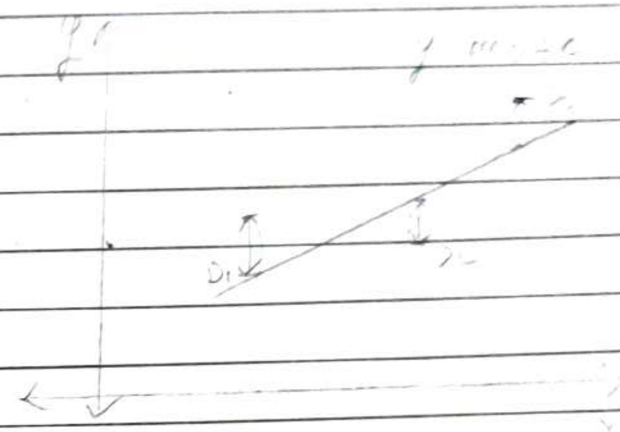
Assignment no 5

* Least Square Technique

1) Straight Line

$$y = mx + c$$

$$a \sum x^2 + b x = \sum xy$$
$$a \sum x + n b = \sum y$$



2) The result of measured electrical resistance (R) of copper bar at various temperature are

Temp	19	25	30	36	40	45	50
Resistance	76	77	79	80	82	83	85

if resistance is related to temp by relation $R = aT + b$ find the value of a & b

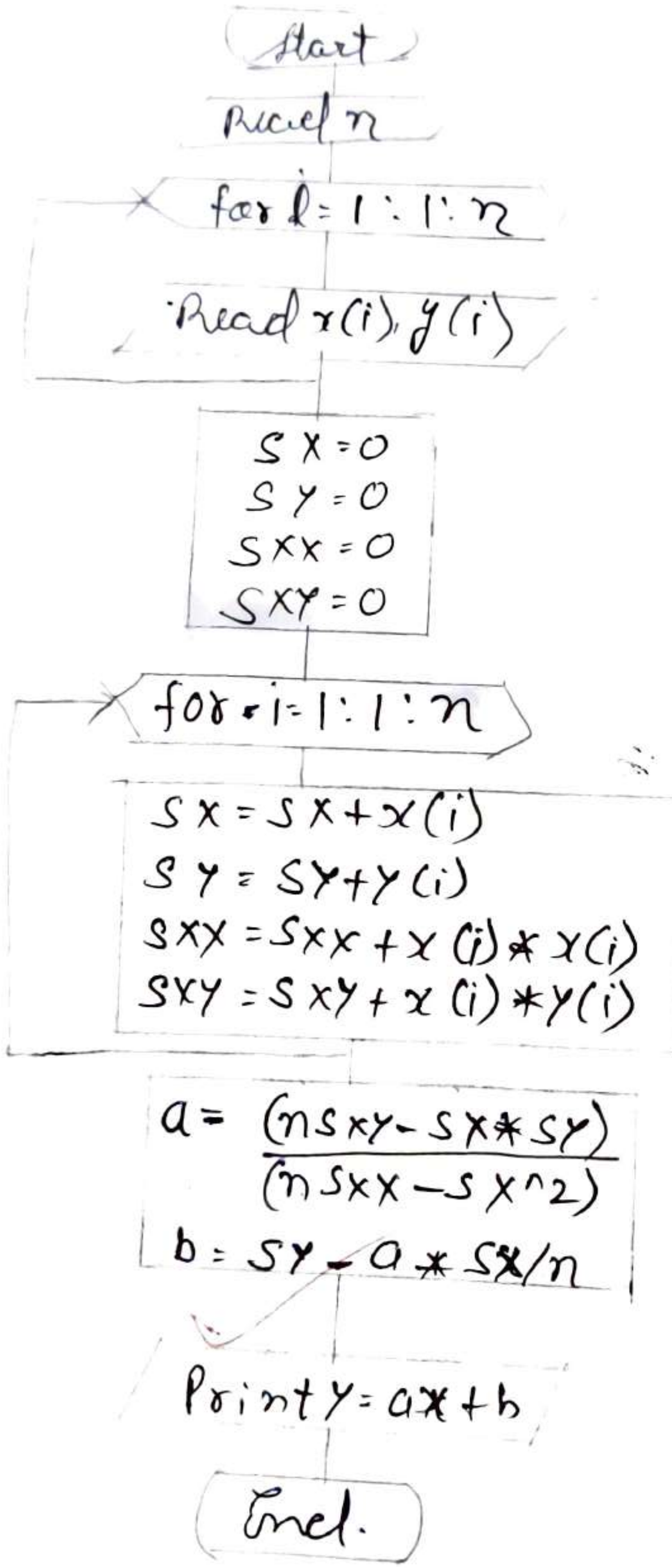
→ $n = 7$

$\sum D^2 = min$

$$a \sum x^2 + b \sum x = \sum xy$$
$$a \sum x + n b = \sum y$$

$\sum T$	$\sum R$	$\sum T^2$	$\sum TR$
19	76	361	1444
25	77	625	1925
30	79	900	2370
36	80	1290	2880
40	82	1600	2380
45	83	2025	3735
50	85	2500	4250
245	562	9307	19884

Flowchart for LST straight line.



```
n = input ('Enter the Number Of intervals
:'); for i= 1:1:n
x(i)= input ('Enter the x value :');
y(i)= input ('Enter the y value :');
end
sx=0;
sy=0;
sxx=0;
sxy=0;
for i= 1:1:n
 sx=sx+x(i);
 sy=sy+y(i);
 sxx=sxx+x(i)*x(i);
 sxy=sxy+x(i)*y(i);
end
a=(n*sxy-sx*sy)/(n*sxx-sx*sx);
b=(sy-a*sx)/n;
fprintf('y=%f*x+%f',a,b);

%OUTPUT
%Enter the Number Of intervals :7
%Enter the x value :19
%Enter the y value :76
%Enter the x value :25
%Enter the y value :77
%Enter the x value :30
%Enter the y value :79
%Enter the x value :36
%Enter the y value :80
%Enter the x value :40
%Enter the y value :82
%Enter the x value :45
%Enter the y value :83
%Enter the x value :50
%Enter the y value :85

y = 0.292350x+70.053474>>
```

Using equation

$$a\sum x^2 + b\sum x = \sum xy$$

$$a\sum x + nb = \sum y$$

$$(a \times 9307) + (b \times 245) = 19884$$

$$(a \times 245) + (b \times 7) = 562$$

by solving the above equation

$a = 0.2923$

$b = 70.05$

$\therefore y = ax + b$

$\therefore y = 0.2923x + 70.05$

2) Quadratic Equation

It is used in fitting quadratic equation to data points in curve fitting

To fit a quadratic equation $y = ax^2 + bx + c$ to a set of data points, you would minimize the sum of the ~~space~~ square of the diff between the observed & predicted value of y for each data point.

$$a\Sigma x^4 + b\Sigma x^3 + c\Sigma x^2 = \Sigma x^2 y$$

$$a\Sigma x^3 + b\Sigma x^2 + c\Sigma x = \Sigma xy$$

$$a\Sigma x^2 + b\Sigma x + nc = \Sigma y$$

Q Solve the value for $y = ax^2 + bx + c$ for given data using quadratic equation method of least square technique

x	1	2	3	4	5	6	7
y	-5	-2	5	16	31	50	73

$n = 7$

$\Sigma D = \min$

x	y	x^4	x^3	x^2	x^2y	xy
1	-5	1	1	1	-5	-5
2	-2	16	8	4	-8	-4
3	5	81	27	9	45	15
4	16	256	64	16	256	64
5	31	625	125	25	775	155
6	50	1296	216	36	1800	300
7	73	2401	343	49	3577	511
28	168	4676	784	140	6440	1036

To find a, b, c

$(a \times 4676) + b(784) + c(140) = 6440$

$(a \times 784) + (b \times 140) + (c \times 28) = 1036$

$(a \times 140) + (b \times 28) + c(7 \times c) = 168$

$\therefore a = 2$

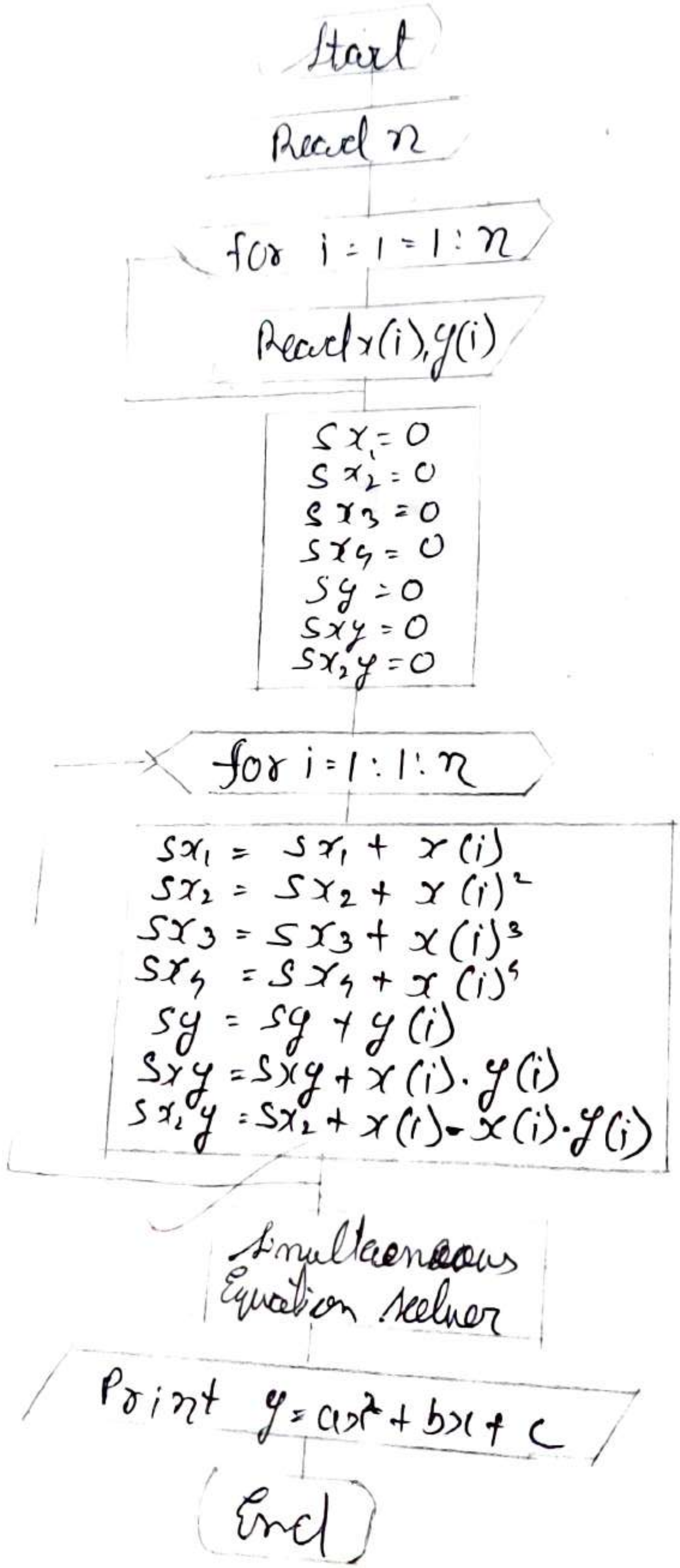
$b = -3$

$c = -4$

$\therefore y = ax^2 + bx + c$

$\therefore y = 2x^2 - 3x - 4$

Flowchart for LST Quadratic equation:



```
% Sarthak Abhay Patil
% TYMEB228
% B4
n = input('Enter the value of n = ');
for i=1:1:n
    x(i)= input('Enter the value of x = ');
    y(i)= input('Enter the value of y = ');
end
Sx = 0;
Sx2 = 0;
Sx3 = 0;
Sx4 = 0;
Sy = 0;
Sxy = 0;
Sx2y = 0;
for i = 1:1:n
    Sx = Sx + x(i);
    Sx2 = Sx2 + x(i)*x(i);
    Sx3 = Sx3 + x(i)^3;
    Sx4 = Sx4 + x(i)^4;
    Sy = Sy + y(i);
    Sxy = Sxy + x(i)*y(i);
    Sx2y = Sx2y + x(i)*x(i)*y(i);
end
    A = [Sx4 Sx3 Sx2; Sx3 Sx2 Sx; Sx2 Sx n];
    B = [Sx2y; Sxy; Sy];

    x = linsolve(A,B);
    a = x(1);
    b = x(2);
    c = x(3);
    fprintf('y = (%f)*x^2 + (%f)*x + %f',a,b,c)

    %OUTPUT
    Enter the value of n = 7
    Enter the value of x = 1
    Enter the value of y = -5
    Enter the value of x = 2
    Enter the value of y = -2
    Enter the value of x = 3
    Enter the value of y = 5
    Enter the value of x = 4
    Enter the value of y = 16
    Enter the value of x = 5
    Enter the value of y = 31
    Enter the value of x = 6
    Enter the value of y = 50
    Enter the value of x = 7
    Enter the value of y = 73
y = (2.000000)*x^2 + (-3.000000)*x + -4.000000>>
```

3) Power curve equation

a) $y = a x^b$

b) $y = a b^x$

c) $y = x y^b$

a) $y = a x^b$

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

$Y = \log y$

$A = b$

$B = \log a$

$X = \log x$

$A \Sigma X^2 + B \Sigma X = \Sigma XY$

$A \Sigma X + nB = \Sigma Y$

$A = b$, $a = e^B$

b) $y = a b^x$

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

$Y = \log y$

$A = \log b$

$X = x$

$B = \log a$

$A \Sigma X^2 + B \Sigma X = \Sigma XY$

$A \Sigma X + nB = \Sigma Y$

$a = e^B$, $b = e^A$

c) $a = x y^b$

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

$\log y = \frac{1}{b} \log a - \frac{1}{b} \log x$

$\therefore Y = \log y$

$X = \log x$

$A = -\frac{1}{b}$

$B = \frac{1}{b}$

$A\Sigma x^2 + B\Sigma x = \Sigma xy$
$A\Sigma x + nB = \Sigma y$
$a = e^b \qquad b = A$

Q) Equation of best fitted curve $y = ax^b$ using least sq criteria for the set of point given below a & b value.

x	0.5	1.5	2	2.5	3
y	0.7425	3.8578	5.9397	8.301	10.912

→ $y = ax^b$

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

$A\Sigma x^2 + B\Sigma X = \Sigma xy$

$A\Sigma x + nB = \Sigma y$

$X = \log x$	$y = \log y$	x^2	xy
-0.693	-0.2977	0.4802	0.2063
0.405	1.3500	0.1640	0.5467
0.693	1.7816	0.4802	1.2346
0.916	2.1163	0.8396	1.9385
1.0986	2.3893	1.2069	2.6245
2.4196	7.3395	3.1707	6.5505

$(A \times 3.1707) + B(2.4196) = 6.5509$

$(A \times 2.4196) + (5 \times B) = 7.3395$

$A = 1.444$

$B = 0.7421$

$A = b = 1.444$

$a = e^{B}$

$= e^{0.7421} = 2.10034$

$\therefore y = ax^{b}$

$y = 2.10034 \times x^{1.444}$

Flowchart for power eqn $y = ax^b$

Start

Read n

for $i = 1:1:n$

Read $x(i)$ $y(i)$

$x(i) = \log x(i)$
$y(i) = \log y(i)$

$SX = 0$ $SXX = 0$
$SY = 0$ $SXY = 0$

for $i = 1:1:n$

$SX = SX + x(i)$
$SY = SY + Y(i)$
$SXX = SXX + x(i)^2$
$SXY = SXY + x(i)Y(i)$

$$A = \frac{n\,SXY - SXSY}{n\,SXX - (SX)^2}$$

$$B = \frac{SY - ASX}{n}$$

$a = \exp(B)$
$b = A$

Print $y = ax^b$

End

```
n = input ('Enter the Number Of intervals :');
for i= 1:1:n
 x(i)= input ('Enter the x value :');
 y(i)= input ('Enter the y value :');
 x(i)=log(x(i));
 y(i)=log(y(i));
end
sx=0;
sy=0;
sxx=0;
sxy=0;
for i= 1:1:n
 sx=sx+x(i);
 sy=sy+y(i);
 sxx=sxx+x(i)*x(i);
 sxy=sxy+x(i)*y(i);
end
A=(n*sxy-sx*sy)/(n*sxx-sx*sx);
B= (sy-A*sx)/n;
a=exp(B);
b=A;
fprintf('\n the values are constant A=%f and B=%f', A,B);
fprintf('\n the equation of power curve is y=%2.4f x^%2.4fx' ,a,b);

%OUTPUT
%Enter the Number Of intervals :5
%Enter the x value :0.5
%Enter the y value :0.7425
%Enter the x value :1.5
%Enter the y value :3.8578
%Enter the x value :2
%Enter the y value :5.9397
%Enter the x value :2.5
%Enter the y value :8.301
%Enter the x value :3
%Enter the y value :10.912

%the values are constant A=1.499974 and B=0.741954
%the equation of power curve is y=2.1000 x^1.5000x>>
```

4) Exponential Curve

$y = a \cdot e^{bx}$

$\log y = \log a + bx$

$Y = \log y$

$A = b$

$X = x$

$B = \log a$

$A \Sigma x^2 + B \Sigma x = \Sigma xy$

$A \Sigma x + n \cdot B = \Sigma y$

$a = e^b \qquad b = A$

Q) Equation of best fitting curve $y = a \cdot e^{bx}$ find a & b

$x =$	2	4	6	8
$y =$	25	38	56	84

$X = x$	$Y = \log y$	X^2	XY
2	3.2189	4	6.4378
4	3.637	16	14.548
6	4.025	36	24.15
8	4.4308	64	35.4464
20	15.3112	120	80.5822

$A(120) + B(10) = 80.5822$

$A(80) + 5(B) = 15.3112$

$\therefore A = 0.20131$

$B = 2.82125$

$b = A = 0.20131$

$a = e^{B} = 16.797$

$y = a \cdot e^{bx}$

$y = 16.79 \cdot e^{0.20131 \cdot x}$

Flowchart for exponential curve.

Start

Read n

for $i = 1:1:n$

Read $x(i)\ y(i)$

$y(i) = \log y(i)$
$x(i) = x(i)$

$Sx = 0 \quad Sxx = 0$
$Sy = 0 \quad Sxy = 0$

for $i = 1:1:n$

$Sx = Sx + x(i)$
$Sy = Sy + y(i)$
$Sxx = Sxx + x(i)^2$
$Sxy = Sxy + x(i)\,y(i)$

$$A = \frac{n\,Sxy - Sx\,Sy}{n\,Sxx - Sx^2}$$

$$B = \frac{Sy - A\,Sx}{n}$$

$a = \exp(B)$
$b = A$

Print $a \cdot e^{xb}$

End.

```
n = input ('Enter the Number Of intervals :');
for i= 1:1:n
 x(i)= input ('Enter the x value :');
 y(i)= input ('Enter the y value :');
 y(i)=log(y(i));
 x(i)=x(i);
end
sx=0;
sy=0;
sxx=0;
sxy=0;
for i= 1:1:n
 sx=sx+x(i);
 sy=sy+y(i);
 sxx=sxx+x(i)*x(i);
 sxy=sxy+x(i)*y(i);
end
A=(n*sxy-sx*sy)/(n*sxx-sx*sx);
B= (sy-A*sx)/n;
a=exp(B);
b=A;
fprintf('\n the values are constant A=%f and B=%f', A,B);
fprintf('\n the equation of expomemtial curve is y=%2.4fe^%2.4fx' ,a,b);

%OUTPUT
%Enter the Number Of intervals :4
%Enter the x value :2
%Enter the y value :25
%Enter the x value :4
%Enter the y value :38
%Enter the x value :6
%Enter the y value :56
%Enter the x value :8
%Enter the y value :84

%the values are constant A=0.201179 and B=2.822261
%the equation of exponential curve is y=16.8148e^0.2012x>>
```

Assignment no 6

* Lagranges Interpolation

Lagrange interpolation is method for finding a polynomial that takes on specific value at certain points it is used in numerical analysis to find the lowest degree polynomial that interpolates a given set of data

The lagrange interpolation theorem can be used to create a polynomial that passes over a group of point & take certain value randomly.

$$x_y = \sum_{i=1}^{n} x_i \prod_{j=1}^{n} \left(\frac{y_g - y_j}{y_i - y_j} \right) \quad i \neq j$$

$$y_y = \sum_{i=1}^{n} y_i \prod_{j=1}^{n} \left(\frac{x_g - x_j}{x_i - x_j} \right) \quad i \neq j$$

$$y_y = f(x_y)$$
$$x_y = f(y_g)$$

a) Using lagranges method of interpolation determine y for $x = 1.1$

x	1	1.2	1.3	1.5
y	1	1.09	1.14	1.22

$\rightarrow$ $x_y = 1.1$

$$y_y = \sum_{i=1}^{n} y_i \prod_{j=1}^{n} \left(\frac{x_y - x_j}{x_i - x_j}\right) \quad i \neq j$$

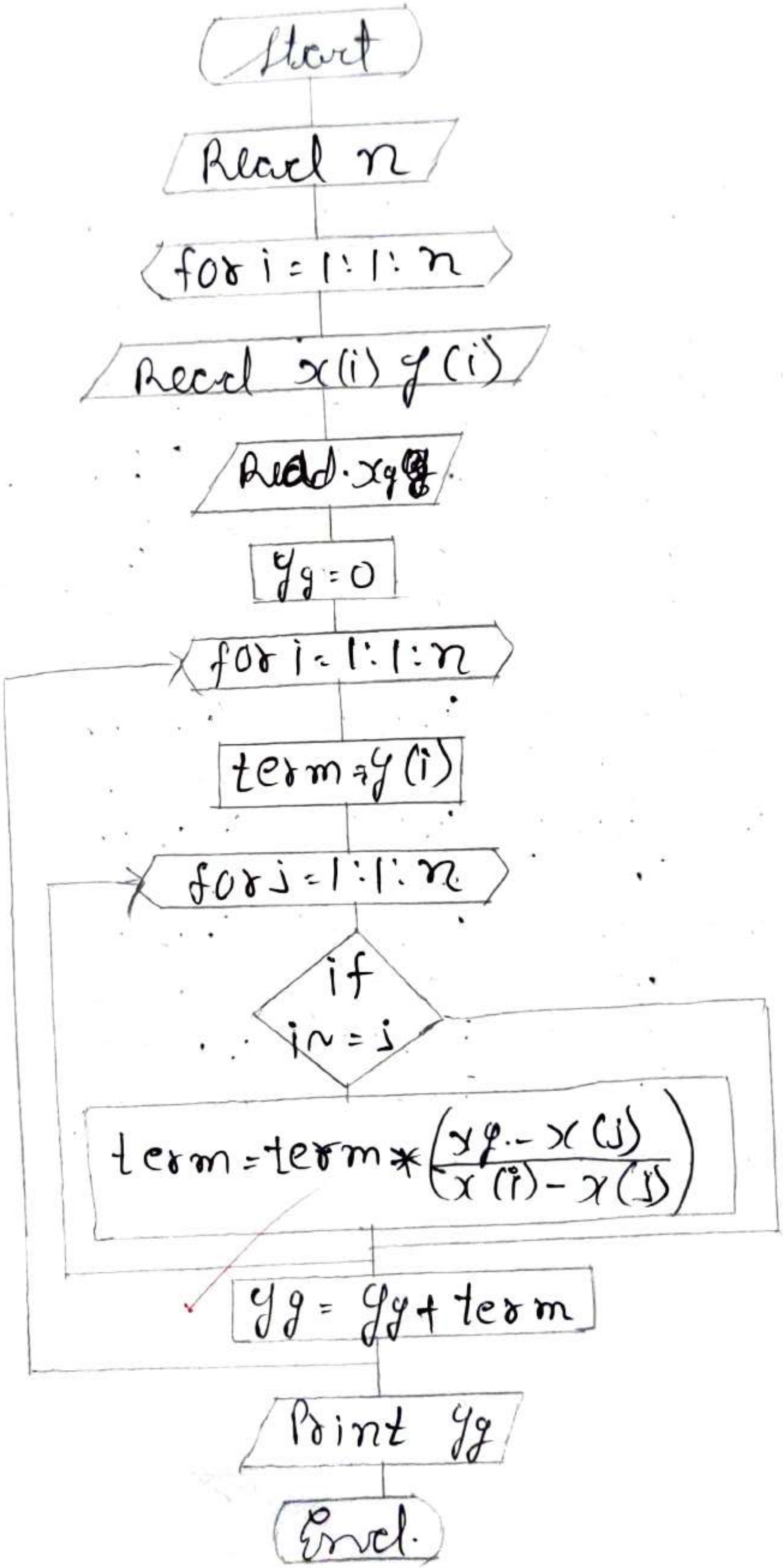
for $n = 4$

$$y_y = y_1\left(\frac{x_y - x_2}{x_1 - x_2}\right)\left(\frac{x_y - x_3}{x_1 - x_3}\right)\left(\frac{x_y - x_4}{x_1 - x_4}\right) + y_2\left(\frac{x_y - x_1}{x_2 - x_1}\right)\left(\frac{x_y - x_3}{x_2 - x_3}\right)\left(\frac{x_y - x_4}{x_2 - x_4}\right) + y_3\left(\frac{x_y - x_1}{x_3 - x_1}\right)\left(\frac{x_y - x_2}{x_3 - x_2}\right)\left(\frac{x_y - x_4}{x_3 - x_4}\right)$$
$$+ y_4\left(\frac{x_y - x_1}{x_4 - x_1}\right)\left(\frac{x_y - x_2}{x_4 - x_2}\right)\left(\frac{x_y - x_3}{x_4 - x_3}\right)$$

$$y_y = 1\left(\frac{1.1 - 1.2}{1 - 1.2}\right)\left(\frac{1.1 - 1.3}{1 - 1.3}\right)\left(\frac{1.1 - 1.5}{1 - 1.5}\right)$$
$$+ 1.09\left(\frac{1.1 - 1}{1.2 - 01}\right)\left(\frac{1.1 - 1.3}{1.2 - 1.3}\right)\left(\frac{1.1 - 1.5}{1.2 - 1.5}\right)$$
$$+ 1.14\left(\frac{1.1 - 1}{1.3 - 1}\right)\left(\frac{1.1 - 1.2}{1.3 - 1.2}\right)\left(\frac{1.1 - 1.5}{1.3 - 1.5}\right)$$
$$+ 1.22\left(\frac{1.1 - 1}{1.5 - 1}\right)\left(\frac{1.1 - 1.2}{1.5 - 1.2}\right)\left(\frac{1.1 - 1.3}{1.5 - 1.3}\right)$$

$$y_y = 0.267 + 1.453 + 0.0813 - 0.76$$
$$= 1.0413$$

Flowchart for Lagranges interpolation



```
n = input('Enter the value of n =');
for i=1:1:n
 x(i) = input('enter value of x');
 y(i) = input('enter value of y');
end
Xg=input('enter Xg');
Yg=0;
for i =1:1:n
 term=y(i);
 for j=1:1:n
 if j~=i
 term=term*(Xg-x(j))/(x(i)-x(j));
end
 end
 Yg=Yg+term
end
fprintf('interpolating value Yg =%f',Yg);
```

```
OUTPUT :
Enter the value of n =4
enter value of x1
enter value of y1
enter value of x1.2
enter value of y1.09
enter value of x1.3
enter value of y1.14
enter value of x1.5
enter value of y1.22
enter Xg1.1

Yg = 0.2667

Yg = 1.7200

Yg = 0.9600

Yg = 1.0413

interpolating value Yg =1.041333>>
```

* Newton forward interpolation:-
It is a numerical method that estimates value of pred function between given data point. It is named after sir Isaac Newton, who contributed to calculus & interpolation is technique of estimating the value of the independent variable

Newton forward interpolation used when strip size is same.

$$y_g = y_1 \frac{(x_g - x_1)}{h} \Delta_1 + \frac{(x_g - x_1)(x_g - x_2)\Delta^2}{2! h^2} +$$

$$\frac{(x_g - x_1)(x_g - x_2)(x_g - x_3)}{3! h^3} \Delta_1^3 + \cdots$$

$x_g + qh$ $\qquad q = \frac{x_g - x_1}{h}$

Q) A experiment is performed to study the failure of steel bar. the applied stress 4 time to fracture are recorded as follow. estimated the probable time to generate fracture where a stress of 15 MPa is applied.

T(mpa)	T(sec)	Δ^1	Δ^2	Δ^3	Δ^4	Δ^5	Δ^6	Δ^7
5	40	-5	2	-3	1	10	-38	94
10	35	-3	-1	-2	11	-28	56	
15	32	-4	-3	9	-17	28		
20	28	-7	6	-8	11			
25	21	-1	-2	3				
30	20	-3	1					
35	17	-2						
40	17							

$x_y = 13$

$h = 5$

$u = \frac{x_y - x_1}{h} = \frac{13.5}{5} = 1.4$

$y_y = y_1 + u\Delta_1^1 + \frac{u(u-1)}{2!}\Delta_1^2 + \frac{u(u-1)(u-2)\Delta_1^3}{3!} +$

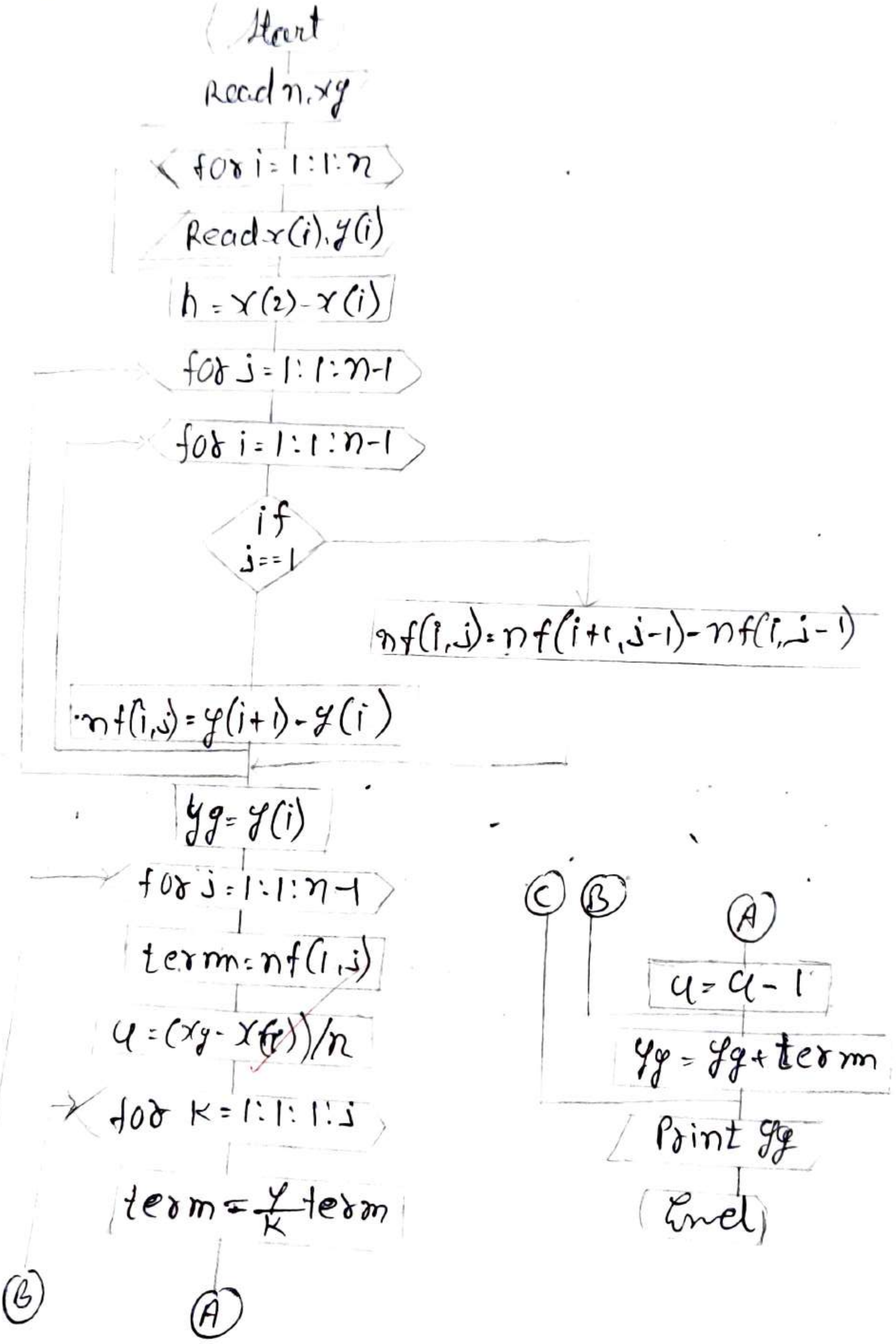
$\frac{u(u-1)(u-2)(u-3)\Delta_1^4}{4!} + \frac{u(u-1)(u-2)(u-3)(u-4)\Delta_1^5}{5!} +$

$\frac{u(u-1)(u-2)(u-3)\Delta^6}{6!} + \frac{u(u-1)(u-2)(u-3)(u-4)(u-5)\Delta_1^6}{6!}$

$\frac{u(u-1)(u-2)(u-3)(u-4)(u-5)(u-6)\Delta_1^7}{7!}$

$= 32.4755$

Flowchart for Newton forward interpolation.



```
% Name- sarthak patil
% Roll no -228
n=input('Enter the number of equation=');
for i=1:1:n
    x(i)=input('Enter the value of x=');
    y(i)=input('Enter the value of y=');
end
xg=input('Enter the value of xg=');
h=x(2)-x(1);
for j=1:1:n-1
    for i=1:1:n-j
        if j==1
            nf(i,j)=y(i+1)-y(i);
        else
            nf(i,j)=nf(i+1,j-1)-nf(i,j-1);
        end
    end
end
yg=y(1);
for j=1:1:n-1
    term=nf(1,j);
    u=(xg-x(1))/h;
    for k=1:1:j
        term=term*u/k;
        u=u-1;
    end
    yg=yg+term
end
fprintf('x(%d)=%f',yg);

% OUTPUT:-
newton_forward_difference
Enter the number of equation=8
Enter the value of x=2
Enter the value of y=19
Enter the value of x=3
Enter the value of y=48
Enter the value of x=4
Enter the value of y=99
Enter the value of x=5
Enter the value of y=178
Enter the value of x=6
Enter the value of y=291
Enter the value of x=7
Enter the value of y=444
Enter the value of x=8
Enter the value of y=643
Enter the value of x=9
Enter the value of y=894
Enter the value of xg=[illegible]
```

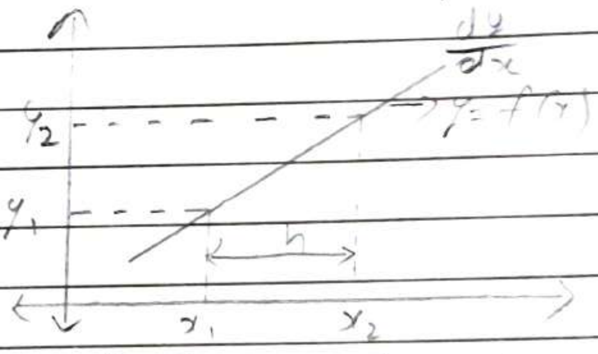
Assignment no 7

Ordinary differential Equation

1) Eulers method :- it is a numerical technique for approximating solution to first order differential equation, it a first order method which means that the local error is proportional to square of the step size. & the global error is proportional to the step size.

The method is based on the idea of approximating a curve using tangent line. The tangent line to a curve at a point is the line that touches the curve at that point & has the same slope as curve at that point

$$\frac{dy}{dx} = \frac{y_2 - y_1}{x_2 - x_1}$$



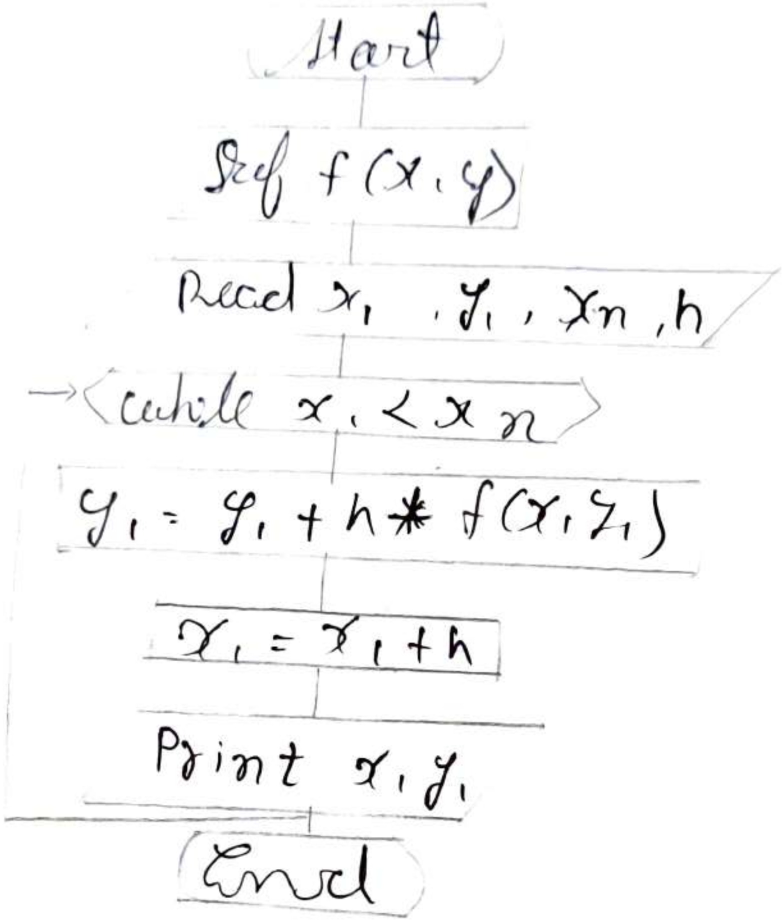
$$y_n = y_{n-1} + h f(x_{n-1}, y_{n-1})$$

2) Given $dy/dx = x^2/(y^2+1)$ with initial condition $y=0$ when $x=0$ obtain y for $x=1$, taking step size (h) as 0.25.

→ $$\frac{dy}{dx} = \frac{x^2}{y^2+1}$$

$x_1 = 0$	$y_1 = 0$
$x_2 = x_1 + h = 0.25$	$y_2 = 0 \times 0.25 \times 0 = 0$
$x_3 = 0.5$	$y_3 = 0 + 0.25 \times 0.0625 = 0.015625$
$x_4 = 0.75$	$y_4 = 0.0781$
$x_5 = 1$	$y_5 = 0.2179$

Flow chart for Eulers method.



```
f=inline('sqrt(x+y)');
x1=input ('Enter the given value of x : ');
y1=input ('Enter the given value of y : ');
xn=input ('Enter value of x for which y to be found : ');
h=input ('Enter increment in x : ');
acc=input('Enter the accuracy=');
while x1<xn
 yp=y1+h*f(x1,y1);
 x2=x1+h;
 yc=y1+(h/2)*(f(x1,y1)+f(x2,yp));
 while abs(yc-yp)>acc
 yp=yc;
 yc=y1+(h/2)*(f(x1,y1)+f(x2,yp));
 end
 x1=x1+h;
 y1=yc;
end
fprintf('\n x=%f,y=%f', x1, y1);
OUTPUT
modifiedeuler
Enter the given value of x : 1
Enter the given value of y : 2.2
Enter value of x for which y to be found : 1.2
Enter increment in x : 0.1
Enter the accuracy=0.001

x=1.200000,y=2.573186>>
```

2) Runge Kutta Method (2nd order) (4th order)

It is numerical method for solving initial value problem of differential equation it can be used to construct high order accurate numerical method without needing high order derivative of funct.

RK 2

$x_2 = x_1 + h$

$k_1 = hf(x_1, y_1)$

$k_2 = hf(x_1 + h, y_1 + k_1)$

$k = \frac{k_1 + k_2}{2}$

$y_2 = y_1 + k$

RK 4

$x_2 = x_1 + h$

$k_1 = hf(x_1, y_1)$

$k_2 = hf(x_1 + h/2, y_1 + k_1/2)$

$k_3 = hf(x_1 + h/2, y_1 + k_2/2)$

$k_4 = hf(x_1 + h, y_1 + k_3)$

$k = \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}$

$y_2 = y_1 + k$

Q) Given DE : ~~$dy/dx = \frac{x^2}{y^2}$~~ $\frac{dy}{dx} = \frac{x^2}{y^2+1}$ with initial condition $y=0$, when $x=0$, obtain Y for $x=1$. (Step size 0

→ $\frac{dy}{dx} = f(x, y) = \frac{x^2}{y^2+1}$

$x_1 = 0 \qquad y_1 = 0$

$x_n = 1 \qquad h = 0.25$

$n = 4$

By RK_2

at $x_2 = x_1 + h = 0.25$

$K_1 = hf(x_1, y_1)$

$= 0.25 \left(\frac{0^2}{0+1}\right)$

$= 0$

$K_2 = hf(x_1 + h, y_1 + K_1)$

$= 0.25\ (0.25, 0)$

$= 0.0156$

$K = \frac{K_1 + K_2}{2} = \frac{0 + 0.0156}{2} = 7.8 \times 10^{-3}$

$y = y_1 + K = 7.8 \times 10^{-3}$

at $x_3 = x_2 + h = 0.5$

$K_1 = 0.25 \left(\frac{0.25^2}{(7.8 \times 10^{-3})^2 + 1}\right)$

$= 0.0156$

$K_2 = 0.25 \left(\frac{0.5^2}{(0.0234)^2 + 1}\right)$

$= 0.0625$

$K = \frac{0.0156 + 0.0625}{2} = 0.0391$

$y_3 = y_2 + K = (7.8 \times 10^{-3}) + (0.0391)$

$= 0.0469$

at $x_4 = x_3 + h = 0.75$

$K_1 = 0.05 \left(\frac{0.5^2}{0.0469^2 + 1}\right)$

$K_2 = 0.25 \left(\frac{0.75}{0.1093^2 + 1}\right)$

$K = \frac{0.0625 + 0.1389}{2} = 0.1007$

$y_4 = y_3 + K = 0.0469 + 0.1007 = 0.1479$

at $x_5 = x_4 + h = 1$

$k_1 = hf(x_4, y_4)$

$= 0.25 \left(\frac{0.75^2}{0.1479^2+1}\right)$ $\quad$ $k_2 = 0.25\left(\frac{1^2}{0.2855^2+1}\right)$

$k_1 = 0.1376$ $\quad$ $k_2 = 0.2312$

$k = \frac{0.1376 + 0.2312}{2} = 0.1844$

$y_5 = y_4 + k$

$= 0.1479 + 0.1844$

$= 0.3324$

$\therefore x = 1 \quad y = 0.3323$

Flowchart:-
Runge Kutta method (RK-2)

Start

Def · 2 = f (x, y)

Read x_1, y_1, x_n, h

While $x_1 < x_n$

$K_1 = h * f(x_1, y_1)$
$K_2 = h * f(x_1 + h, y_1 + k_1)$
$K_3 = \frac{K_1 + K_2}{2}$
$y_1 = y_1 + K$

$x_1 = x_1 + h$

Print = x_1, y_1

End

```
z=inline('sqrt(x+y)');
x1=input ('Enter the given value of x : ');
y1=input ('Enter the given value of y : ');
xn=input ('Enter value of x for which y to be found : ');
h=input ('Enter increment in x : ');
while x1<xn
 k1=h*f(x1,y1);
 k2=h*f(x1+h,y1+k1);
 k=(k1+k2)/2;
 y1+y1+k;
 x1=x1+h;
end
fprintf('x1=%f,y1=%f',x1,y1)
OUTPUT
Enter the given value of x : 1
Enter the given value of y : 2.2
Enter value of x for which y to be found : 1.2
Enter increment in x : 0.1
x1=1.200000,y1=2.200000>>
```

Q Apply Runge Kutta method to find an approximate value of $x = 0.2$ given $dy/dx = x + y$ & $y = 1$ when, $x = 0$ take $h = 0.2$

→

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

$x_1 = 0$ $y_1 = 0$

$x_n = 0.2$ $h = 0.2$

$n = 1$

at $x_2 = x_1 + h = 0.2$

$k_1 = 0.2 (0 + 1)$
$= 0.2$

$k_2 = 0.2 (0.1 + 1.1)$
$= 0.24$

$k_3 = 0.2 (0.1 + 1.12)$
$= 0.244$

$k_4 = 0.2 (0.2 + 1.244)$
$= 0.2888$

$$K = \frac{k_1 + (k_2 + k_3)2 + k_4}{6}$$

$$= \frac{0.2 + 2 \times 0.24 + 2 \times 0.244 + 0.2888}{6}$$

$= 0.2428$

$y_2 = y_1 + K = 1 + 0.2428 = 1.2428$

$\therefore x = 0.2, y = 1.2428$

Runge Kutta method (RK-4)

Start

Def $x = f(x, y)$

Read x_1, y_1 x_n h

while $x_1 < x_n$

$k_1 = h * f(x_1, y_1)$

$k_2 = h * f(x_1 + h/2, y_1 + \frac{k_1}{2})$

$k_3 = h * f(x_1 + h/2, y_1 + k_2/2)$

$k_4 = h * f(x_1\ h, y_1 + k_3)$

$K = 1/6 (k_1 + 2k_2 + 2k_3 + k_4)$

$y_1 = y_1 + K$

$X_1 = X_1 + h$

Print x_1 y_1

End.

```
z=inline('sqrt(x+y)');
x1=input ('Enter the given value of x : ');
y1=input ('Enter the given value of y : ');
xn=input ('Enter value of x for which y to be found : ');
h=input ('Enter increment in x : ');
while x1<xn
 k1=h*f(x1,y1);
 k2=h*f(x1+h/2,y1+(k1)/2);
 k3=h*f(x1+h/2,y1+(k2)/2);
 k4=h*f(x1+h,y1+k3);
 k=(1/6)*(k1+2*k2+2*k3+k4);
 y1=y1+k;
 x1=x1+h;
end
fprintf('\n x1=%f,y1=%f',x1,y1)
```

OUTPUT

```
Enter the given value of x : 1
Enter the given value of y : 2.2
Enter value of x for which y to be found : 1.2
Enter increment in x : 0.1

x1=1.200000,y1=2.573206
```

3) Simultaneous equation

$\frac{dy}{dx} = f(x, y, z)$ $\frac{dz}{dx} = g(x, y, z)$

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

RK-2

$KY_1 = hf(x, y, z)$

$KY_2 = hf(x_1 + h, Y_1 + KY_1, Z + KZ_1)$

$KY = \frac{KY_1 + KY_2}{2}$

$Y_n = Y_{n-1} + KY$

$KZ_1 = hg(x_1, y_1, z_1)$

$KZ_2 = hg(x_1 + h, y_1 + KY_1, z_1 + KZ_1)$

$KZ = \frac{KZ_1 + KZ_2}{2}$

$Z_n = Z_{n-1} + KZ$

Q) solve the foll simultaneous eqn $y' = yz$ & $z' = xy$ find $y(0.1)$ & $z(0.2)$ given that the value of $y(0.0) = 1$ & $z(0.0) = 1$

→ $y' = \frac{dy}{dx} = yz = f(x, y, z)$

$z' = \frac{dz}{dx} = xy = g(x, y, z)$

$h = 0.2 - 0.1 = 0.1$

at $x_2 = x_1 + h = 0.1$

$KY_1 = hf(x_1, y_1, z_1)$

$= 0.1(1 \times 1)$

$= 0.1$

$KZ_1 = hg(x_1, y_1, z_1)$

$= 0.1(0 \times 1)$

$= 0$

$KY_2 = hf(x_1 + h, y_1 + KY_1, z_1 + KZ_1)$

$= 0.1(0.1, 1.1, 1)$

$= 0.11$

$KZ_2 = hg(x_1 + h, y_1 + KY_1, z_1 + K_4)$

$= 0.1(0.1 \times 1.1)$

$= 0.011$

$K_Y = \frac{KY_1 + KY_2}{2}$

$= \frac{0.1 + 0.11}{2}$

$= 0.105$

$K_Z = \frac{KZ_1 + KZ_2}{2}$

$= \frac{0 + 0.011}{2}$

$= 0.0055$

$Y_n = Y_{n-1} + K_Y$

$= 1 + 0.105$

$= 1.105$

$Z_n = Z_{n-1} + K_Z$

$= 1 + 0.0055$

$= 1.0055$

at $x_3 = x_2 + h = 0.2$

$KY_1 = hf(x_2, y_2, z_2)$

$= 0.1(0.105 \times 1.0055)$

$= 0.1111$

$KZ_1 = hg(x_2, y_2, z_2)$

$= 0.1(0.1 \times 1.105)$

$= 0.0111$

$Ky_2 = hf(x_2 + h, y_2 + ky, z_2 + kz_2)$
$= 0.1\ (1.2161 \times 1.066)$
$= 0.1236$

$Kz_2 = hf\ (x_2 + h_0, y_2 + k_{21}, z_2 + K_{21})$
$= 0.1\ (0.2 \times 1.2161)$
$= 0.243$

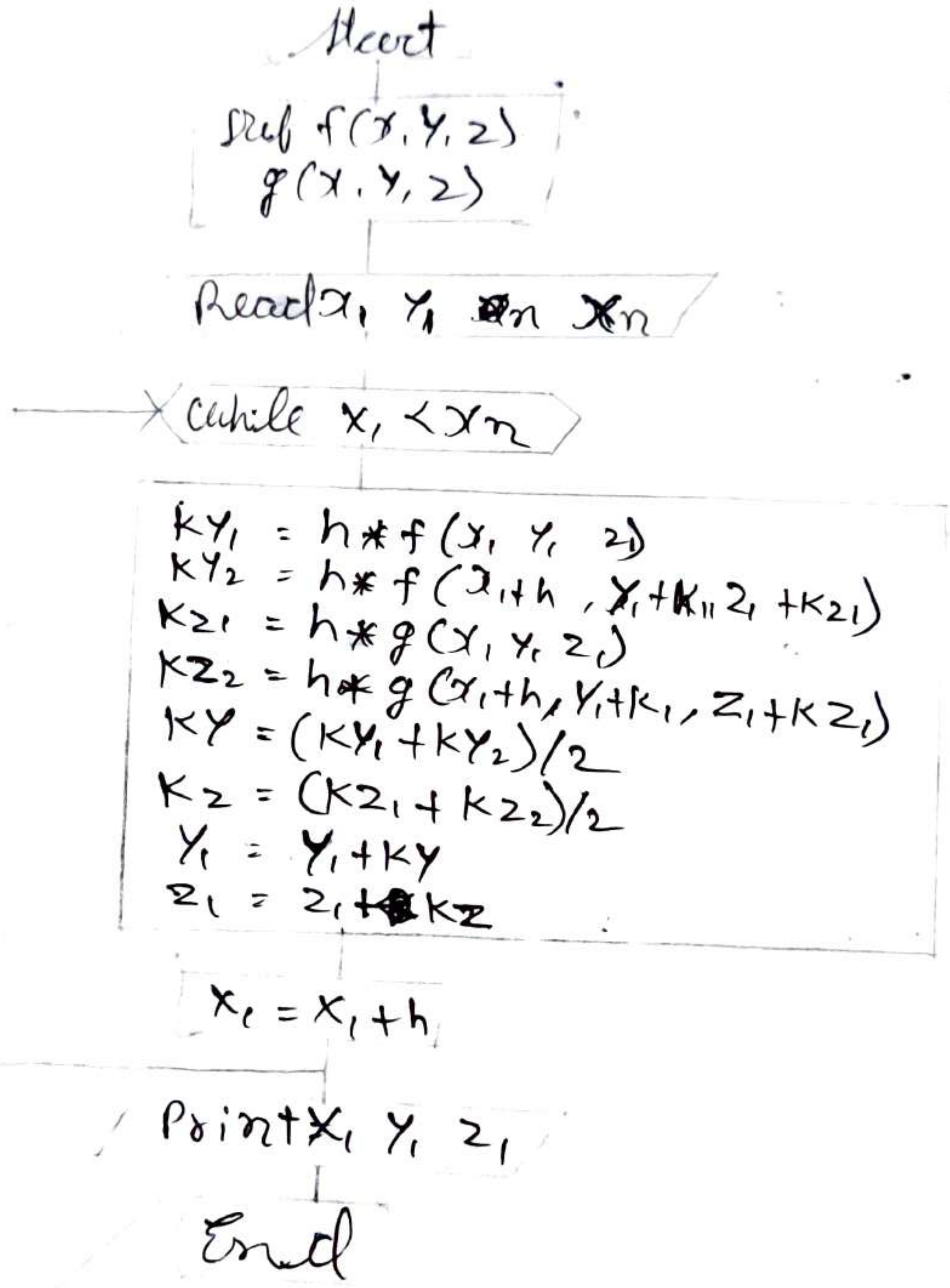
$Ky = \dfrac{ky_1 + ky_2}{2}$
$= \dfrac{0.111 + 0.1236}{2}$
$= 0.11735$

$K_2 = \dfrac{Kz_1 + Kz_2}{2}$
$= \dfrac{0.0111 + 0.0243}{2}$
$= 0.0177$

$y_3 = y_2 + ky$
$= 1.105 + 0.11735$
$= 1.2224$

$y_4 = z_2 + kz$
$= 1.0055 + 0.0177$
$= 1.0232$

Flowchart for simultaneous RK2



```
%NAME: Sarthak Abhay Patil
%DIV:B
%BATCH:B-4
% SimultaneousRK2Method

function sim_rk1_methjod()
x1=input('Enter the vlaue of x1:');
y1=input('Enter the value of y1:');
z1=input('Enter the evalue of z1:');
h=input('Enter stepsize h:');
xn=input('Enter xn value:');
while x1<xn
    ky1=h*f(x1,y1,z1);
    kz1=h*g(x1,y1,z1);
    ky2=h*f(x1+h,y1+ky1,z1+kz1);
    kz2=h*g(x1+h,y1+ky1,z1+kz1);
    ky=(ky1+ky2)/2;
    kz=(kz1+kz2)/2;
    y1=y1+ky;
    z1=z1+kz;
    x1=x1+h;
    fprintf('\n%f    %f    %f',x1,y1,z1);
end
fprintf('\nat xn=%f, yn=%f',x1,y1,z1);
end
function s=f(x,y,z)
s=y*z;
end
function w=g(x,y,z)
w=x*y
end

% OUTPUT
% Enter the vlaue of x1:0
% Enter the value of y1:1
% Enter the evalue of z1:1
% Enter stepsize h:0.1
% Enter xn value:0.2
%
% w =
%
%      0
%
%
% w =
%
%      0.1100
%
%
% 0.100000    1.105000    [illegible]
% w =
```

Assignment no 08

Laplace Equation

$$A\frac{\partial^2 x}{\partial x^2} + \frac{B\partial^2 u}{\partial x \partial y} + \frac{C\partial^2 u}{\partial y^2} + \frac{D\partial u}{\partial x} + \frac{E\partial u}{\partial y} + Fu = 0$$

$s<0$ Elliptical equation

Laplace eqn

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

Poissons eqn

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y)$$

$s=0$ Parabolic equation

Schmidt equation

$$\frac{a\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial y}$$

$s>0$ Hyperbolic equation

wave equation

$$\frac{\partial^2 u}{\partial x^2} = \frac{a\partial^2 u}{\partial y^2}$$

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

$$\frac{\partial^2 u}{\partial y^2} = \frac{u(i,j+1) - 2u(i,j) + u(i,j-1)}{k^2}$$

$h=k$

$$4u(i,j) = u(i+1),j) + u(i-1,j) + u(i,j+1)$$
$$+ u(i,j-1).$$

$u(i,j) = \frac{1}{4}[u(i+1, j) + u(i-1, j) + u(i, j+1) + u(i, j-1)]$

Q $\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$ Solve the laplace eqn with given grid. Find T_1, T_2, T_3 & T_4.

→ $u(i,j) = \frac{1}{4}[u(i+1, j) + u(i-1, j) + u(i, j+1) + u(i, j-1)]$

$T_1 = \frac{1}{4}[40 + T_2 + 60 + T_4]$

$= \frac{1}{4}[100 + T_2 + T_4]$ ---- (1)

$T_2 = \frac{1}{4}[T_1 + 50 + 60 + T_3]$

$= \frac{1}{4}[110 + T_1 + T_3]$ ---- (2)

$T_3 = \frac{1}{4}[60 + T_4 + T_2 + 20]$

$= \frac{1}{4}[60 + T_4 + T_2]$ ---- (3)

$T_4 = \frac{1}{4}[30 + T_3 + T_1]$ ---- (4)

Using Gauss Seidel

acc = 0.01

$T_1 = T_2 = T_3 = T_4 = 0$

$T_1 = 25$

$T_2 = 33.75$

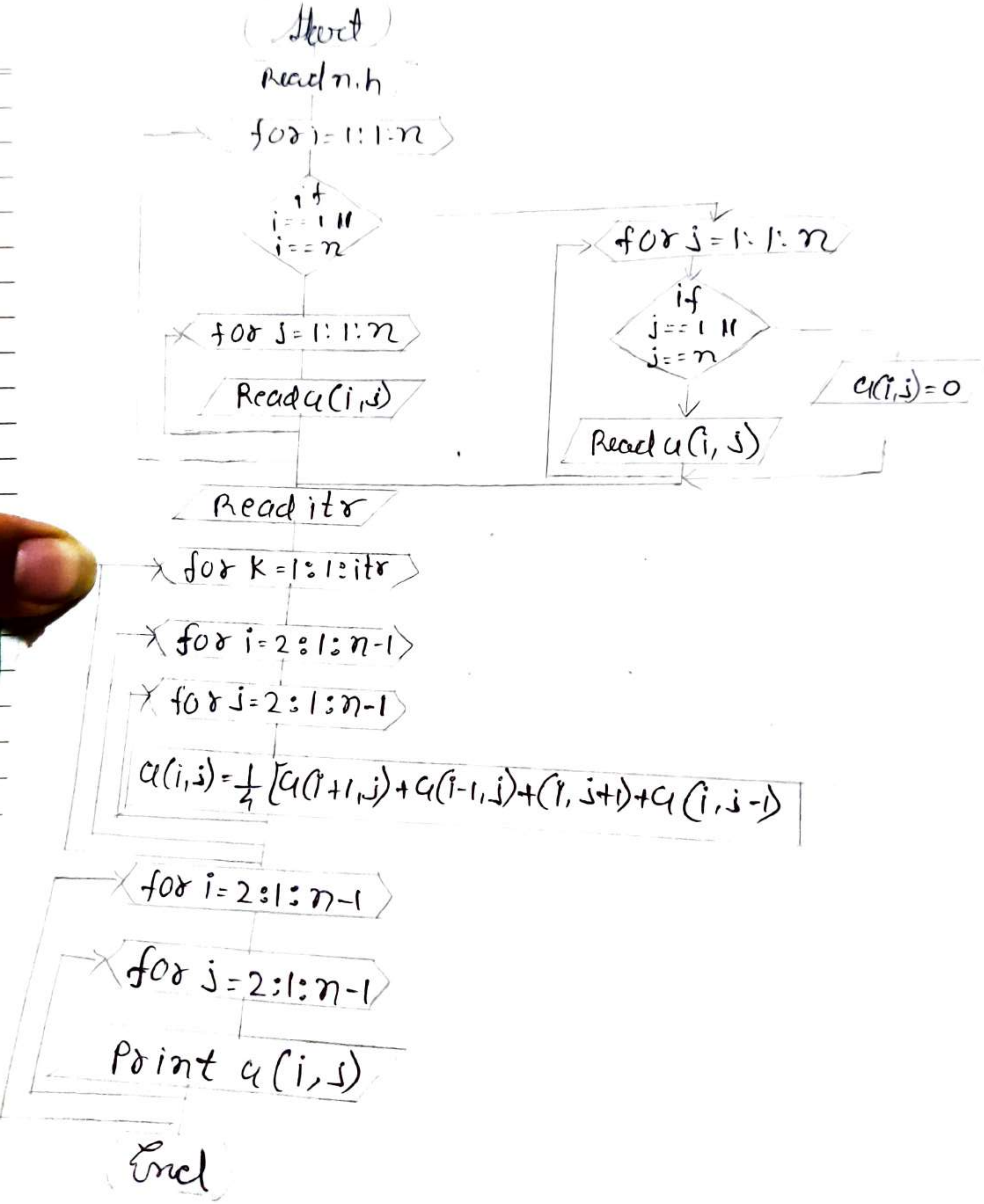
$T_3 = 23.4375$

$T_4 = 19.6093$

by substituting value $T_1 = T_2 = T_3 = T_4 = 0$ in (1)

itr	T_1	T_2	T_3	T_4
1	0	0	0	0
2	25	33.75	23.437	19.609
3	33.339	41.644	30.325	23.4164
4	41.277	45.400	32.204	25.8704
5	42.817	46.255	33.0314	26.4623
6	43.1794	46.5527	33.2537	26.6082
7	43.2902	46.6359	33.3110	26.6503
8	43.3219	46.6581	33.3271	26.6621

Flowchart Laplace



```
n=input("enter the value of n");
%h=input("Enter the value of h:');
for i=1:1:n
 if i==1||i==n
 for j=1:1:n
 u(i,j)=input("Enter the value of u(i,j)=");
 end
 else
for j=1:1:n
 if j==1||j==n
 u(i,j)=input("Enter the value of u(i,j)");
 else
 u(i,j)=0;
 end
 end
 end
end
itr=input("Enter the number of iterations=");
for k=1:1:itr
 for i=2:1:n-1
 for j=2:1:n-1
 u(i,j)=(1/4)*(u(i+1,j)+u(i-1,j)+u(i,j+1)+u(i,j-1)); end
 end
end
for i=2:1:n-1
 for j=2:1:n-1
 fprintf('%f\n',u(i,j));
 end
end


%output
 Laplace
enter the value of n4
Enter the value of u(i,j)=60
Enter the value of u(i,j)=40
Enter the value of u(i,j)=20
Enter the value of u(i,j)=0
Enter the value of u(i,j)60
Enter the value of u(i,j)10
Enter the value of u(i,j)60
Enter the value of u(i,j)20
Enter the value of u(i,j)=60
Enter the value of u(i,j)=50
Enter the value of u(i,j)=40
Enter the value of u(i,j)=30
Enter the number of iterations=15
43.333333
26.666667
```

[TYMFB228]

ASSIGNMENT No. 9

Title : Matlab Solver for all Numerical Methods

1. Roots of Equation

```
>> f = inline('x^2 + 10*x -3100')

f =

    Inline function:
    f(x) = x^2 + 10*x -3100

>> root = fzero(f,0)

root =

  50.9017
```

2. Numerical Integration

```
>> f = inline('3*(0.4+0.004*T)')

f =

    Inline function:
    f(T) = 3*(0.4+0.004*T)

>> I = quad(f,25,125)

I =

  210
```

3. Simultaneous Equation

```
>> A = [2 4 3;3 6 1;1 3 2]

A =

     2     4     3
     3     6     1
     1     3     2

>> B = [13;16;9]

B =

    13
    16
     9

>> X = linsolve(A,B)

X =

    1.0000
    2.0000
    1.0000
```

4. Curve Fitting

a) Straight line

>> **X = [19 25 30 36 40 45 50]**

X =

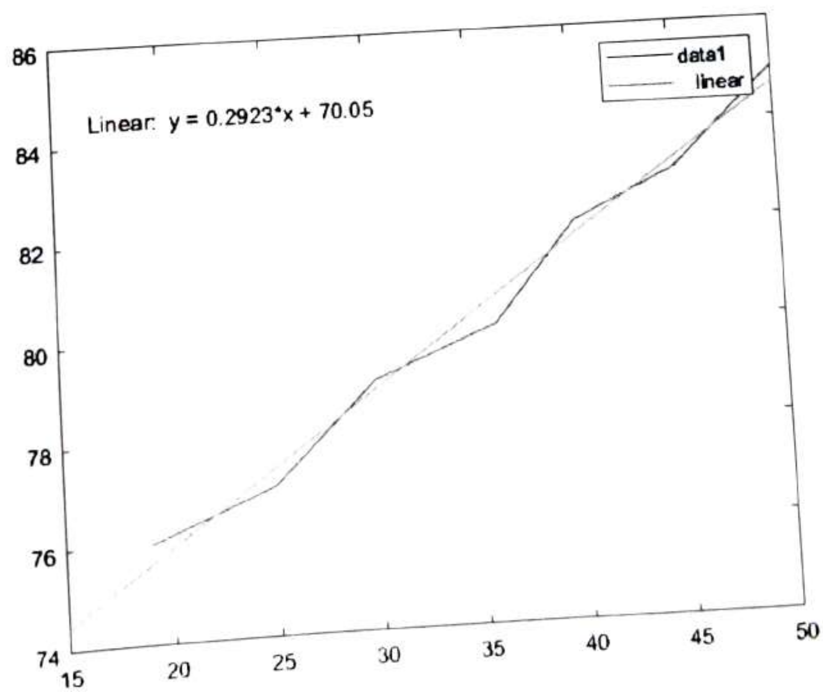
19 25 30 36 40 45 50

>> **Y = [76 77 79 80 82 83 85]**

Y =

76 77 79 80 82 83 85

>> **plot(X,Y)**



(Answer : Linear: y = 0.2923*x + 70.05)

5. **Interpolation**

```
>> X = [-12 10 38]

X =

  -12   10   38

>> Y = [50.1 10 4.9]

Y =

  50.1000  10.0000   4.9000

>> Xg = interp1(Y,X,16,"spline")

Xg =

 -15.6600
```

6. **Ordinary Diff. Equation**

```
>> f = inline('(x^2)/(y^2 + 1)')

f =

     Inline function:
     f(x,y) = (x^2)/(y^2 + 1)

>> [Xn Yn] = ode23(f,[0,1],0)
```

Xn =	Yn =
0	0
0.0250	0.0000
0.0500	0.0000
0.0731	0.0001
0.0962	0.0003
0.1192	0.0006
0.1423	0.0010
0.1654	0.0015
0.1918	0.0024
0.2225	0.0037
0.2581	0.0057
0.2995	0.0090
0.3474	0.0140
0.4030	0.0218
0.4677	0.0341
0.5431	0.0533
0.6315	0.0838
0.7315	0.1298
0.8315	0.1894
0.9315	0.2634
1.0000	0.3222