INDEX

CONTENTS

MODULE 1

- Least Squares Method
- Bisection Method
- Regula Falsie Method
- Newton Raphson Method
- Gauss – Elimination Method
- Gauss – Jordon Method
- Gauss – Seidel Method

MODULE 2

- Newton's Gregory Forward Method
- Newton's Gregory Backward Method
- Lagrange's Method
- Newton's Divided Difference Method

MODULE 3

- Newton's Forward Difference Method
- Newton's Backward Difference Method
- Trapezoidal Method
- Simpson's $1/3^{th}$ Method
- Simpson's $3/8^{th}$ Method

MODULE 4

- Taylor's Method
- Euler Method
- Modified Euler Method
- Improved Euler Method
- Runge Kutta 2^{nd} Order Method
- Runge Kutta 3^{rd} Order Method
- Runge Kutta 4^{th} Order Method
- Milne's Method
- Adams Bash Forth Method

MODULE 5

- Liebman's Method
- Poisson's Method
- Blender – Schmidt Method
- Crank – Nicolson Method
- Explicit Scheme Method

MODULE

1

Exercise: 01

LEAST SQUARE METHOD

Objective:

Fitting the given data line/curve using least square method

Input:

The Country and its percentage of NEET Exam passing in year 2019.

Country	Passing Rate
1	52
2	71
3	44
4	58
5	75
6	67
7	36
8	70
9	59
10	46
11	39

Link:

https://news.careers360.com/neet-2019-result-statistics-released-nta-797042-candidates-qualified-exam/amp

Procedure/Methodology:

Country (x)	1	2	3	4	5	6	7	8	9	10	11
Percentage of Passing (y)	52	71	44	58	75	67	36	70	59	46	39

Tabular Column:

X	Y	U= X-6	V= Y-67	U^2	U^3	U^4	U^2V	UV
1	52	-5	-15	25	-125	625	-375	75
2	71	-4	4	16	-64	256	64	16
3	44	-3	-23	9	-27	81	-207	69
4	58	-2	-9	4	-8	16	-36	18
5	75	-1	8	1	-1	1	8	-8
6	67	0	0	0	0	0	0	0
7	36	1	-31	1	1	1	-31	-31
8	70	2	3	4	8	16	12	6
9	59	3	-8	9	27	81	-72	-24
10	46	4	-21	16	64	256	-336	-82
11	39	5	-28	25	125	625	-700	-140
		ΣU= 0	**ΣV= -120**	**ΣU^2= 110**	**ΣU^3= 0**	**ΣU^4= 1958**	**$\Sigma U^2 V$= -1673**	**ΣUV= -133**

<u>Straight line:</u>

$$\sum \boldsymbol{v} = \boldsymbol{a}\boldsymbol{\Sigma}\boldsymbol{u} + \boldsymbol{b}\boldsymbol{n}$$

$$\sum \boldsymbol{u}\boldsymbol{v} = \boldsymbol{a}\boldsymbol{\Sigma}\boldsymbol{u}^{\mathbf{2}} + \boldsymbol{b}\boldsymbol{\Sigma}\boldsymbol{u}$$

⇒

-120=0(a) +11(b) -- (1)

-133=110(a) +0(b) -- (2)

From Equation (1)

-120=11b

$$b = \frac{-120}{11}$$

$$\boxed{b = -10.9091}$$

From Equation (2)

-133=110a

a = $\frac{-133}{110}$

$\boxed{a = -1.2091}$

Substitute the value of a & b,

Where v = Y-67 and u = X-6

$$\boldsymbol{v = au + b}$$

v = -1.21(u) + (-10.91)

v = -1.21(X-6) + (-10.91)

v = -1.21 (X) + 7.26 – 10.91

Y – 67 = -1.21 X – 3.65

Y = -1.21 X + 63.35

$\boxed{\mathbf{1.21\ X+Y = 63.35}}$

<u>Curve fitting</u>

$$\boldsymbol{\sum v = a\sum u^2 + b\Sigma u + cn}$$

$$\boldsymbol{\sum uv = a\Sigma u^3 + b\Sigma u^2 + c\sum u}$$

$$\boldsymbol{\sum u^2 v = a\sum u^4 + b\Sigma u^3 + c\sum u^2}$$

$\Rightarrow$

-120 = 110(a) + 0(b) + 11(c) -- (1)

-133 = 0(a) + 110(b) + 0(c) -- (2)

-1673 = 1958(a) + 0(b) + 110(c) -- (3)

From Equation (2)

-133 = 0(a) + 110(b) + 0(c)

-133 = 110(b)

b = -133 / 110

$\boxed{b = -1.21}$

By Solving Equation (1) and (3)

(1) x 10		1100 (a) + 110 $\cancel{(c)}$ = -1200
(3) x 1	(-)	1958 (a) + $\cancel{110}$ (c) = -1673
		-858 (a) = 473

a = 473 / - 858

$\boxed{a = -0.5513}$

Substitute the value a = -0.5513 in equation (1)

110 (a) + 11 (c) = -120

110(-0.5513) + 11 (c) = -120

11 (c) = -120 + 60.643

c = -59.357/11

$\boxed{c = -5.357}$

Substitute the value of a, b & c

Where u = X-6 and v = -1.21 (X) – 3.65

$$\boldsymbol{v = au^2 + bu + c}$$

V = (-0.5513) u^2 + (-1.21) + (-5.357)

-1.21 X – 3.65 = (-0.5513) $(x-6)^2$ + (-1.21) $(x-6)$ - 5.357

-1.21 X – 3.65 = (-0.5513) (x^2-12x +6) + (-1.21) (x-6) - 5.357

$\cancel{-1.21 X}$ – 3.65 = -0.5513 x^2 + 6.6156 x – 19.8468 – $\cancel{1.21 x}$ + 7.26 – 5.357

-3.65 = -0.5513x^2 + 6.6156 x -17.9829

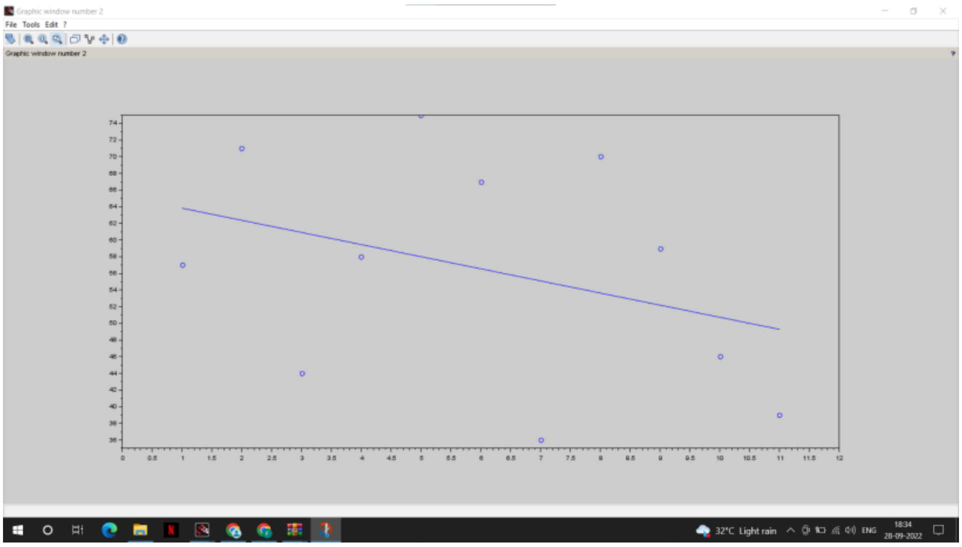
$\boxed{Y = 0.5513x^2 – 6.6156 x + 14.3329}$

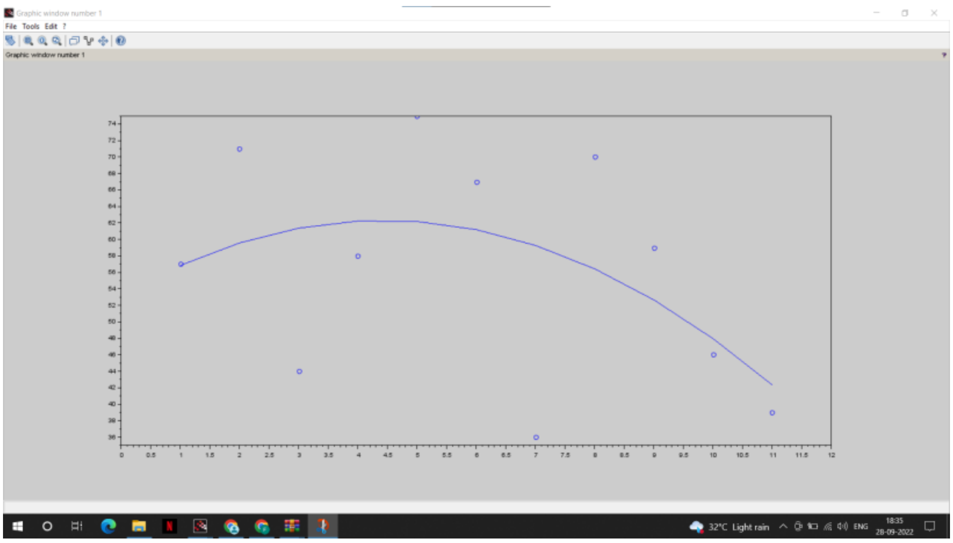
Coding:

```
clc; clear all;
x=[1:1:11];
y=[52,71,44,58,75,67,36,70,59,46,39];
n=length(x);

sx=sum(x);
sy=sum(y);
sx4=sum(x.^4);
sx3=sum(x.^3);
sx2=sum(x.^2);
sxy=sum(x.*y);
sx2y=sum((x.^2).*y);
//
disp("Curve Fitting")
//sy=a1*sx2+b1*sx+c1*n;
//sxy=a1*sx3+b1*sx2+c1*sx;
//sx2y=a1*sx4+b1*sx3+c1*sx2;

A=[sx2,sx,n;sx3,sx2,sx;sx4,sx3,sx2];
B=[sy,sxy,sx2y]';
//X=[a1,b1,c1]';
R1=inv(A);
X=R1*B;
disp(A,'A=')
disp(B,'B=')
disp(R1,'R1=')
a1=X(1); b1=X(2); c1=X(3);
disp(a1,'a1=')
disp(b1,'b1=')
disp(c1,'c1=')
printf("Equation : y = (%.4f).x^2 + (%.4f).x +(%.4f)",a1,b1,c1);

figure(1);
y1=a1*x^2+b1*x+c1
scatter(x,y)
plot(x,y1)
//
disp("Line Fitting")
//sy=a2*sx+b2*n;
//sxy=a2*sx2+b2*sx;

A1=[sx,n;sx2,sx];
B1=[sy,sxy]';
//X=[a2,b2]';
R2=inv(A1);
X1=R2*B1;
disp(A1,'A1=')
disp(B1,'B1=')
```

```
disp(R2,'R2=')

a2=X1(1); b2=X1(2);
disp(a2,'a2=')
disp(b2,'b2=')
printf("Equation : y = (%.4f).x + (%.4f)",a2,b2);
figure(2);
y2=a2*x+b2
scatter(x,y)
plot(x,y2)
```

Output:





Exercise: 2

BISECTION METHOD

Objective:

Fitting the root using Bisection method

Input:

$3x + \sin x - e^x = 0$

Link:

http://sdcamzn.in/Images/DisclosureFile/CBNSTASSIGN1.pdf

Procedure/Methodology:

$f(x) = 3x + \sin(x) - e^x = 0$

When x = 0

$f(0) = -1$ (-ve)

When x = 1

$f(1) = 1.1232$ (+ve)

Iteration 1:

Here $f(0) = -1 < 0$ &

$f(1) = 1.1232 > 0$

∴ Root lies between 0 & 1

$X_0 = \frac{0+1}{2}$

$\boxed{X_0 = 0.5}$

$f(X_o) = f(0.5) = 3(0.5) + \sin(0.5) - e^{0.5}$

$= 0.3307 > 0$

Iteration 2:

Here $f(0) = -1 < 0$ &

$f(0.5) = 0.3307 > 0$

∴ Root lies between 0 & 0.5

$X_1 = \frac{0+0.5}{2}$

$\boxed{X_1 = 0.25}$

$f(X_1) = f(0.25) = 3(0.25) + \sin(0.25) - e^{0.25}$

$= -0.2866 < 0$

Iteration 3:

Here $f(0.25) = -0.2866 < 0$ &

$f(0.5) = 0.3307 > 0$

∴ Root lies between 0.25 & 0.5

$X_2 = \frac{0.25+0.5}{2}$

$\boxed{X_2 = 0.375}$

$f(x_2) = f(0.375) = 3(0.375) + \sin(0.375) - e^{0.375}$

$= 0.0363 > 0$

Iteration 4:

Here $f(0.25) = -0.2866 < 0$ &

$f(0.375) = 0.0363 > 0$

∴ Root lies between 0.25 & 0.375

$X_3 = \frac{0.25+0.375}{2}$

$\boxed{X_3 = 0.3125}$

$f(x_3) = f(0.3125) = 3(0.3125) + \sin(0.3125) - e^{0.3125}$

$= -0.1219 < 0$

Iteration 5:

Here $f(0.3125) = -0.1219 < 0$ &

$f(0.375) = 0.0363 > 0$

∴ Root lies between 0.3125 & 0.375

$X_4 = \frac{0.3125+0.375}{2}$

$\boxed{X_4 = 0.3438}$

$f(x_4) = f(0.3438) = 3(0.3438) + \sin(0.3438) - e^{0.3438}$

$= -0.042 < 0$

Iteration 6:

Here $f(0.3438) = -0.042 < 0$ &

$f(0.375) = 0.0363 > 0$

$\therefore$ Root lies between 0.3438 & 0.375

$X_5 = \frac{0.3438+0.375}{2}$

$\boxed{X_5 = 0.3594}$

$f(x_5) = f(0.3594) = 3(0.3594) + \sin(0.3594) - e^{0.3594}$

$= -0.0026 < 0$

Iteration 7:

Here $f(0.3594) = -0.0026 < 0$ &

$f(0.375) = 0.0363 > 0$

$\therefore$ Root lies between 0.3594 & 0.375

$X_6 = \frac{0.3594+0.375}{2}$

$\boxed{X_6 = 0.3672}$

$f(x_6) = f(0.3672) = 3(0.3672) + \sin(0.3672) - e^{0.3672}$

$= 0.0169 > 0$

Iteration 8:

Here $f(0.3672) = 0.0169 > 0$ &

$f(0.3594) = -0.0026 < 0$

$\therefore$ Root lies between 0.3672 & 0.3594

$X_7 = \frac{0.3672+0.3594}{2}$

$\boxed{X_7 = 0.3633}$

$f(x_7) = f(0.3633) = 3(0.3633) + \sin(0.3633) - e^{0.3633}$

$= 0.0071 > 0$

Iteration 9:

Here $f(0.3633) = 0.0071 > 0$ &

$f(0.3594) = -0.0026 < 0$

$\therefore$ Root lies between 0.3633 & 0.3594

$X_8 = \frac{0.3633+0.3594}{2}$

$\boxed{X_8 = 0.3613}$

$f(x_8) = f(0.3613) = 3(0.3613) + \sin(0.3613) - e^{0.3613}$

$= 0.0023 > 0$

Iteration 10:

Here $f(0.3613) = 0.0023 > 0$ &

$f(0.3594) = -0.0026 > 0$

$\therefore$ Root lies between 0.3613 & 0.3594

$X_9 = \frac{0.3613+0.3594}{2}$

$\boxed{X_9 = 0.3604}$

$f(x_9) = f(0.3604) = 3(0.3604) + \sin(0.3604) - e^{0.3604}$

$= -0.0002 > 0$

Result:

The approximation root of the equation $3x + \sin x - e^x = 0$ using bisection method is 0.3604 (after 10 iterations)

Coding:

```
clc;clear all;
deff('y=f(x)','y=3*x+sin(x)-%e^x');
x1=input("Enter first approximation: ");
x2=input("Enter second approximation : ");
d=input("Enter accuracy : ");
c=1;
printf('Succesive approximation by Bisection Method : \n\t   x1\t   \tx2\t   \tm\t   \tf(m)\n');
while abs(x1-x2)>d
   m=(x1+x2)/2;
   printf('\t%f\t%f\t%f\t%f\n',x1,x2,m,f(m));
   if f(m)*f(x1)>0
      x1=m;
   else
      x2=m;
   end
   c=c+1;
end
printf('\nNumber of Iterations : %d\n\n',c);
printf('The solution of equation after %i iteration is %g',c,m);
```

Output:

Scilab 6.1.1 Console

File Edit Control Applications ?

Scilab 6.1.1 Console

```
Enter first approximation: 0

Enter second approximation : 1

Enter accuracy : 0.0001

Succesive approximation by Bisection Method :
           x1              x2              m               f(m)
        0.000000        1.000000        0.500000        0.330704
        0.000000        0.500000        0.250000        -0.286621
        0.250000        0.500000        0.375000        0.036281
        0.250000        0.375000        0.312500        -0.121899
        0.312500        0.375000        0.343750        -0.041956
        0.343750        0.375000        0.359375        -0.002620
        0.359375        0.375000        0.367188        0.016886
        0.359375        0.367188        0.363281        0.007147
        0.359375        0.363281        0.361328        0.002267
        0.359375        0.361328        0.360352        -0.000175
        0.360352        0.361328        0.360840        0.001046
        0.360352        0.360840        0.360596        0.000435
        0.360352        0.360596        0.360474        0.000130
        0.360352        0.360474        0.360413        -0.000023

Number of Iterations : 15

The solution of equation after 15 iteration is 0.360413
-->
```

Exercise: 3

REGULAR FALSI METHOD

Objective:

Find the roots using Regular Falsi method

Input:

$3x + \sin x - e^x = 0$

Link:

http://sdcamzn.in/Images/DisclosureFile/CBNSTASSIGN1.pdf

Procedure/Methodology:

$f(x) = 3(x) + \sin x - e^x$

When x = 0

$f(0) = -1$ (-ve)

When x = 1

$f(1) = 1.1232$ (+ve)

Iteration 1:

Here $f(0) = -1 < 0$ &

$f(1) = 1.1232 > 0$

∴ Root lies between 0 & 1

Here $x_0=0$ & $x_1 = 1$

$$x_2 = \frac{x0.f(x1)-x1.f(x0)}{f(x_1)-f(x_0)}$$

$$= \frac{0(1.1232)-1(-1)}{1.1232-(-1)}$$

$$\boxed{x_2 = 0.471}$$

$f(x_2) = f(0.471) = 3(0.417) + \sin(0.417) - e^{0.417}$

$= 0.2652 > 0$

Iteration 2:

Here $f(0.471) = 0.2652 > 0$ &

$f(0) = -1 < 0$

∴ Root lies between 0.471 & 0

Here $x_0 = 0$ & $x_1 = 0.471$

$$x_3 = \frac{x0.f(x1) - x1.f(x0)}{f(x_1) - f(x_0)}$$

$$= \frac{0(0.2652) - 0.471(-1)}{0.2652 - (-1)}$$

$$\boxed{x_3 = 0.3723}$$

$f(x_3) = f(0.3723) = 3(0.3723) + \sin(0.3723) - e^{0.3723}$

$= 0.0295 > 0$

Iteration 3:

Here $f(0.3723) = 0.0295 > 0$ &

$f(0) = -1 < 0$

$\therefore$ Root lies between 0 & 0.3723

Here $x_0=0$ & $x_1 = 0.3723$

$$x_4 = \frac{x0.f(x1) - x1.f(x0)}{f(x_1) - f(x_0)}$$

$$= \frac{0(0.0295) - 0.3723(1)}{0.0295 - (-1)}$$

$$\boxed{x_4 = 0.3616}$$

$f(x_4) = f(0.3616) = 3(0.3616) + \sin(0.3616) - e^{0.3616}$

$= 0.0029 > 0$

Iteration 4:

Here $f(0.3616) = 0.0029 > 0$ &

$f(0) = -1 < 0$

$\therefore$ Root lies between 0.3616 & 0

Here $x_0= 0$ & $x_1 = 0.3616$

$$x_5 = \frac{x0.f(x1) - x1.f(x0)}{f(x_1) - f(x_0)}$$

$$= \frac{0(0.0029) - 0.3616(-1)}{0.0029 - (-1)}$$

$$\boxed{x_5 = 0.3605}$$

$f(x_5) = f(0.3605) = 3(0.3605) + \sin(0.3605) - e^{0.3605}$

$= 0.0003 > 0$

Iteration 5:

Here $f(0.3605) = 0.0003 > 0$ &

$f(0) = -1 < 0$

∴ Root lies between 0.3605 & 0

Here $x_0 = 0$ & $x_1 = 0.3605$

$$X_6 = \frac{x0.f(x1) - x1.f(x0)}{f(x_1) - f(x_0)}$$

$$= \frac{0(0.0003) - 0.3605\,(-1)}{0.0003 - (-1)}$$

$$\boxed{X_6 = 0.3604}$$

$f(x_6) = f(0.3604) = 3(0.3604) + \sin(0.3604) - e^{0.3604}$

$= 0.0003 > 0$

Iteration 6:

Here $f(0.3604) = 0.0003 > 0$ &

$f(0) = -1 < 0$

∴ Root lies between 0.3604 & 0

Here $x_0 = 0$ & $x_1 = 0.3604$

$$X_7 = \frac{x0.f(x1) - x1.f(x0)}{f(x_1) - f(x_0)}$$

$$= \frac{0(0.0003) - 0.3604\,(-1)}{0.0003 - (-1)}$$

$$\boxed{X_7 = 0.3604}$$

$f(x_6) = f(0.3604) = 3(0.3604) + \sin(0.3604) - e^{0.3604}$

$= 0.0003 > 0$

Result:

The approximation root of the equation $3x + \sin x - e^x = 0$ using Regula falsi method is 3.604 (after 6 iterations)

Coding:

```
clc; clear all;
deff('y=f(x)','y=3*x+sin(x)-%e^x');
a=input("Enter first approximation: ");
b=input("Enter second approximation : ");
i=1;
printf('Successive Approximation by Regula Falsi Method : \n\t a\t  \tb\t  \tc\t  \tf(c)\n\n');
while(i<=6)
   c=(a*f(b)-b*f(a))/(f(b)-f(a));
   printf('\t%f\t%f\t%f\t%f\n',a,b,c,f(c));
   if (f(a)*f(c)<0) then
      b=c;
   else
      a=c;
   end
   i=i+1;
end
//disp(i)
printf('\nNumber of Iterations : %d\n\n',i);
printf('The solution of equation after 7 iterations is %g\n\n',c);
```

Output:

```
Scilab 6.1.1 Console
File  Edit  Control  Applications  ?
Scilab 6.1.1 Console

Enter first approximation: 0

Enter second approximation : 1

Successive Approximation by Regula Falsi Method :
         a               b               c               f(c)

        0.000000        1.000000        0.470990        0.265159
        0.000000        0.470990        0.372277        0.029534
        0.000000        0.372277        0.361598        0.002941
        0.000000        0.361598        0.360537        0.000289
        0.000000        0.360537        0.360433        0.000028
        0.000000        0.360433        0.360423        0.000003

Number of Iterations : 7

The solution of equation after 7 iterations is 0.360423


-->
```

Exercise: 4

NEWTON RAPHSON METHOD

Objective:

Find the roots using Newton Raphson method

Input:

$3x + \sin(x) - e^x = 0$

Procedure/Methodology:

$f(x) = 3(x) + \sin(x) - e^x$

$f'(x) = 3 + \cos(x) - e^x$

When x = 0

$f(0) = -1$ (-ve)

When x = 1

$f(1) = 1.1232$ (+ve)

Here $f(0) = -1 < 0$ &

$f(1) = 0.839 > 0$

∴Root lies between 0 & 1

$x_0 = \frac{0+1}{2} = 0.5$

$\boxed{x_0 = 0.5}$

Iteration 1:

$f(x_0) = f(0.5) = 3(0.5) + \sin(0.5) - e^{0.5}$

$= 0.3307$

$f'(x_0) = f'(0.5) = 3(0.5) + \cos(0.5) - e^{0.5}$

$= 1.4$

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

$x_1 = 0.5 - \frac{0.3307}{1.4}$

$\boxed{x_1 = 0.3516}$

Iteration 2 :

$f(x_1) = f(0.3516) = 3(0.3516) + \sin(0.3516) – e^{0.3516}$

$= - 0.0221$

$f'(x_1) = f'(0.3516) = 3 + \cos(0.3516) - e^{0.3516}$

$= 2.5174$

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

$x_2 = 0.3516 – \frac{(-0.0221)}{2.5174}$

$\boxed{x_2 = 0.3604}$

Iteration 3:

$f(x_2) = f(0.3604) = 3(0.3604) + \sin(0.3604) - e^{0.3604}$

$= 0$

$f'(x_2) = f'(0.3604) = 3 + \cos(0.3604) - e^{0.3604}$

$=2.5019$

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

$x_3 = 0.3604 - \frac{0}{2.5019}$

$\boxed{x_3 = 0.3604}$

Result:

The Approximation roots of the equation $3x + \sin x - e^x = 0$ using Newton Raphson method is 3.604 (after 3 iterations)

Coding:

```
clc; clear all;
deff('y=f(x)','y=3*x+sin(x)-%e^x');
deff('z=f1(x)','z=3+cos(x)-%e^x');
a=input("Enter first approximation: ");
b=input("Enter second approximation : ");
x=(a+b)/2;
i=1;
printf('Successive approximation by Newton Raphson Method:\n\t  x\t  \tf(x)\t  \tf1(x)\n');
while(i<=4)
   x=(x-(f(x)/f1(x)));
   printf("\t%f\t%f\t%f\n",x,f(x),f1(x));
   i=i+1
end
printf('\n Number of Iterations :%d\n',i);
printf('\nThe Solution of Equation after %d iteration is %f',i,x);
```

Output:

```
Scilab 6.1.1 Console
File  Edit  Control  Applications  ?
Scilab 6.1.1 Console

Enter first approximation: 0

Enter second approximation : 1

Successive approximation by Newton Raphson Method:
          x               f(x)            f1(x)
        0.351626        -0.022073       2.517436
        0.360394        -0.000068       2.501863
        0.360422        -0.000000       2.501814
        0.360422        0.000000        2.501814

 Number of Iterations :5

The Solution of Equation after 5 iteration is 0.360422
-->
```

Exercise:5

GAUSS ELIMINATION METHOD

Objective:

Find the solution of system of linear equations using Gauss elimination method.

Input:

$$a + b + 3c - d = 4$$
$$3b - c + d = 2$$
$$a + 2b - 2c + 5d = 0$$
$$a + b - 5c + 2d = 3$$

Link:

https://images.app.goo.gl/FxnGBUk3qCdoyhpw8

Procedure/Methodology:

$$a + b + 3c - d = 4$$
$$3b - c + d = 2$$
$$a + 2b - 2c + 5d = 0$$
$$a + b - 5c + 2d = 3$$

Converting the equations into matrix form, AX=B

$$\begin{array}{ccc} A & X & & B \\ \begin{pmatrix} 1 & 1 & 3 & -1 \\ 0 & 3 & -1 & 4 \\ 1 & 2 & -2 & 5 \\ 1 & 1 & -5 & 2 \end{pmatrix} & \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} & = & \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \end{array}$$

Augmented Matrix:

$$[A, B] = \left(\begin{array}{cccc|c} 1 & 1 & 3 & -1 & 4 \\ 0 & 3 & -1 & 4 & 2 \\ 1 & 2 & -2 & 5 & 0 \\ 1 & 1 & -5 & 2 & 3 \end{array}\right)$$

$R3 \leftarrow R3 - R1$

$$= \left(\begin{array}{cccc|c} 1 & 1 & 3 & -1 & 4 \\ 0 & 3 & -1 & 4 & 2 \\ 0 & 1 & -5 & 6 & -4 \\ 1 & 1 & -5 & 2 & 3 \end{array}\right)$$

$R4 \leftarrow R4 - R1$

$$= \left(\begin{array}{cccc|c} 1 & 1 & 3 & -1 & 4 \\ 0 & 3 & -1 & 4 & 2 \\ 0 & 1 & -5 & 6 & -4 \\ 0 & 0 & -8 & 3 & -1 \end{array}\right)$$

$R3 \leftarrow R3 - 0.3333 \times R2$

$$= \left(\begin{array}{cccc|c} 1 & 1 & 3 & -1 & 4 \\ 0 & 3 & -1 & 4 & 2 \\ 0 & 0 & -4.67 & 4.67 & -4.67 \\ 0 & 0 & -8 & 3 & -1 \end{array}\right)$$

$R4 \leftarrow R4 - 1.7143 \times R3$

$$= \left(\begin{array}{cccc|c} 1 & 1 & 3 & -1 & 4 \\ 0 & 3 & -1 & 4 & 2 \\ 0 & 0 & -4.67 & 4.67 & -4.67 \\ 0 & 0 & 0 & -5 & 7 \end{array}\right)$$

We get,

$$a + b + 3c - d = 4 \quad \longrightarrow (1)$$

$$3b - c + 4d = 2 \quad \longrightarrow (2)$$

$$-4.67\, c + 4.67\, d = -4.67 \quad \longrightarrow (3)$$

$$-5d = 7 \quad \longrightarrow (4)$$

Now use back substitution method,

From equation (4)

$$-5d = 7$$

$$\boxed{d = -1.4}$$

In equation (3), Substitute the value of d,

$$-4.67\ c + 4.67\ d = -4.67$$

$$-4.67\ c + 4.67\ (-1.4) = -4.67$$

$$-4.67\ c = 1.868$$

$$\boxed{c = -0.4}$$

In equation (2), Substitute the value of d & c,

$$3b - c + 4d = 2$$

$$3b - (-0.4) + 4(-1.4) = 2$$

$$3b - 5.2 = 2$$

$$\boxed{b = 2.4}$$

In equation (1), Substitute the value of b c & d,

$$a + b + 3c - d = 4$$

$$a + (2.4) + 3(-0.4) - (1.4) = 4$$

$$a + 2.6 = 4$$

$$\boxed{a = 1.4}$$

Result:

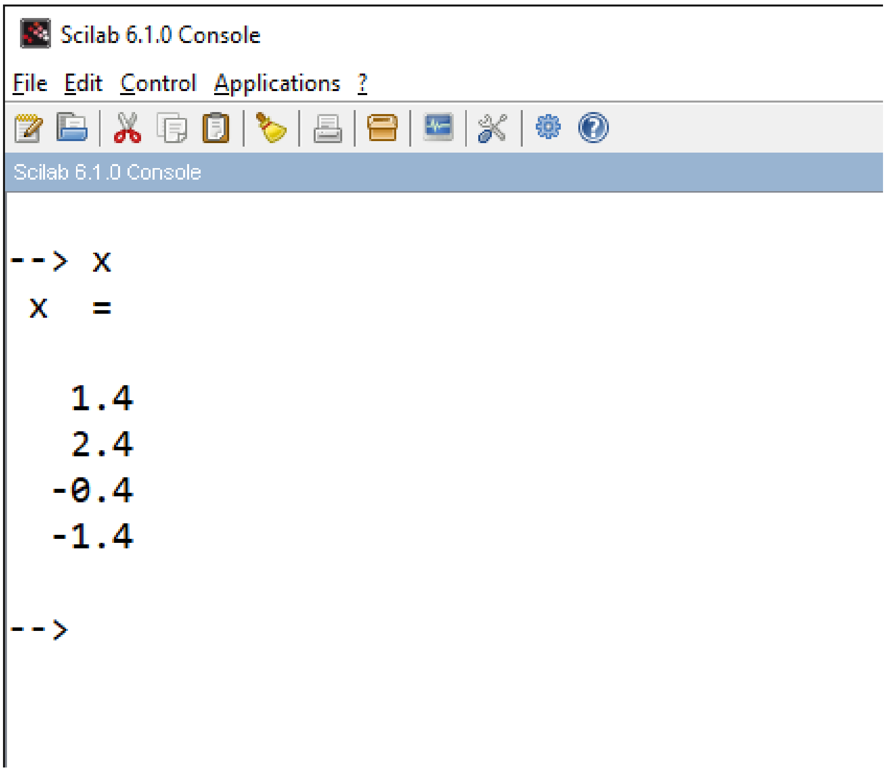
The solution of system of linear equation using Gauss Elimination Method is

a = 1.4; b = 2.4; c = -0.4; d = 1.4

Coding:

```
\\elimination
C =[1,1,3,-1;0,3,-1,4;1,2,-2,5;1,1,-5,2];
b= [4,2,0,3]'
A = [C b];
n= size(A,1);
x = zeros(n,1);
for i=1:n-1
    for j=i+1:n
        m = A(j,i)/A(i,i)
        A(j,:) = A(j,:) - m*A(i,:)
    end
end
x(n) = A(n,n+1)/A(n,n)
for i=n-1:-1:1
    su = 0
    for j=i+1:n
        su = su + A(i,j)*x(j)
        x(i) = (A(i,n+1) - su)/A(i,i)
    end
end
```

Output:



Exercise: 6

GAUSS JORDON METHOD

Objective:

Find the solution of system of linear equations using Gauss Jordon Method.

Input:

$$a + b + 3c - d = 4$$
$$3b - c + d = 2$$
$$a + 2b - 2c + 5d = 0$$
$$a + b - 5c + 2d = 3$$

Link:

https://images.app.goo.gl/FxnGBUk3qCdoyhpw8

Procedure/Methodology:

$$a + b + 3c - d = 4$$
$$3b - c + d = 2$$
$$a + 2b - 2c + 5d = 0$$
$$a + b - 5c + 2d = 3$$

Converting the equations into matrix form,AX=B

$$\underset{A}{\begin{pmatrix} 1 & 1 & 3 & -1 \\ 0 & 3 & -1 & 4 \\ 1 & 2 & -2 & 5 \\ 1 & 1 & -5 & 2 \end{pmatrix}} \underset{X}{\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}} = \underset{B}{\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}}$$

Augmented Matrix:

$$[A, B]= \left(\begin{array}{cccc|c} 1 & 1 & 3 & -1 & 4 \\ 0 & 3 & -1 & 4 & 2 \\ 1 & 2 & -2 & 5 & 0 \\ 1 & 1 & -5 & 2 & 3 \end{array}\right)$$

$R3 \leftarrow R3 - R1$

$$= \left(\begin{array}{cccc|c} 1 & 1 & 3 & -1 & 4 \\ 0 & 3 & -1 & 4 & 2 \\ 0 & 1 & -5 & 6 & -4 \\ 1 & 1 & -5 & 2 & 3 \end{array}\right)$$

$R4 \leftarrow R4 - R1$

$$= \left(\begin{array}{cccc|c} 1 & 1 & 3 & -1 & 4 \\ 0 & 3 & -1 & 4 & 2 \\ 0 & 1 & -5 & 6 & -4 \\ 0 & 0 & -8 & 3 & -1 \end{array}\right)$$

$R2 \leftarrow R2 \div 3$

$$= \begin{pmatrix} 1 & 1 & 3 & -1 & 4 \\ 0 & 1 & -0.3333 & 1.3333 & 0.6667 \\ 0 & 1 & -5 & 6 & -4 \\ 0 & 0 & -8 & 3 & -1 \end{pmatrix}$$

$R1 \leftarrow R1 - R2$

$$= \begin{pmatrix} 1 & 0 & 3.3333 & -2.3333 & 3.3333 \\ 0 & 1 & -0.3333 & 1.3333 & 0.6667 \\ 0 & 1 & -5 & 6 & -4 \\ 0 & 0 & -8 & 3 & -1 \end{pmatrix}$$

$R3 \leftarrow R3 - R2$

$$= \begin{pmatrix} 1 & 0 & 3.3333 & -2.3333 & 3.3333 \\ 0 & 1 & -0.3333 & 1.3333 & 0.6667 \\ 0 & 0 & -4.6667 & 4.6667 & -4.6667 \\ 0 & 0 & -8 & 3 & -1 \end{pmatrix}$$

$R3 \leftarrow R3 \times -0.2143$

$$= \begin{pmatrix} 1 & 0 & 3.3333 & -2.3333 & 3.3333 \\ 0 & 1 & -0.3333 & 1.3333 & 0.6667 \\ 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & -8 & 3 & -1 \end{pmatrix}$$

$R1 \leftarrow R1 - 3.3333 \times R3$

$$= \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & -0.3333 & 1.3333 & 0.6667 \\ 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & -8 & 3 & -1 \end{pmatrix}$$

$R2 \leftarrow R2 + 0.3333 \times R3$

$$= \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & -8 & 3 & -1 \end{pmatrix}$$

$R4 \leftarrow R4 + 8 \times R3$

$$= \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & -5 & 7 \end{pmatrix}$$

We Get,

a = 1.4

b = 2.4

c = - 0.4

d = - 1.4

Result:

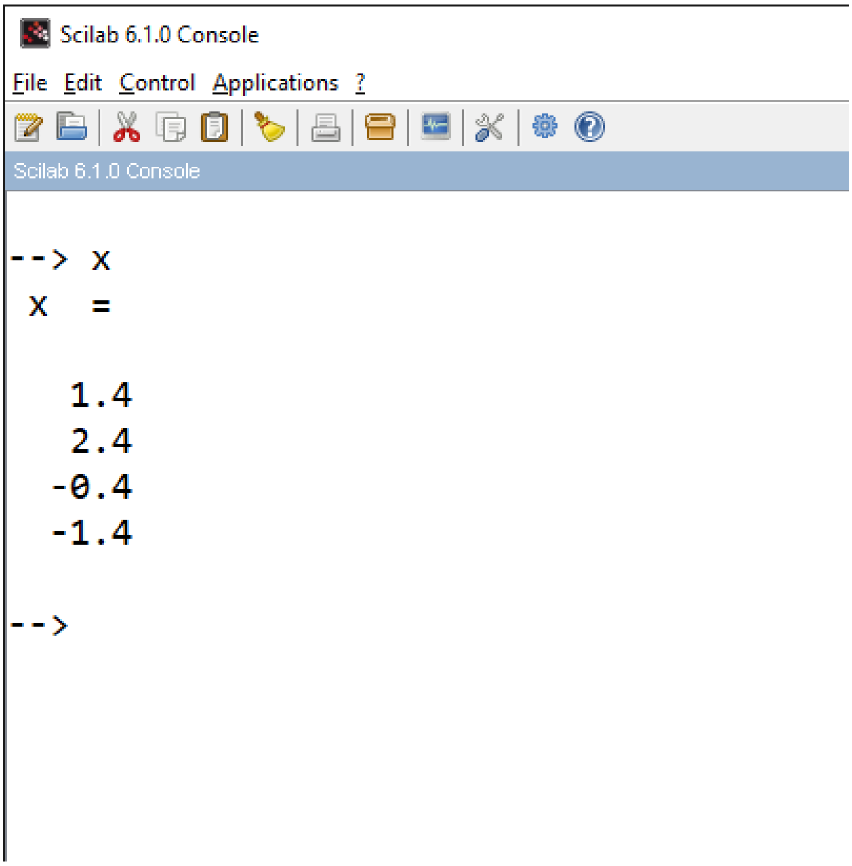
The solution of system of linear equation using Gauss Jordon Method is

a = 1.4; b = 2.4; c = -0.4; d = 1.4

Coding:

```
//jordan
clc; clear all;
C =[1,1,3,-1;0,3,-1,4;1,2,-2,5;1,1,-5,2];
b= [4,2,0,3]'
A = [C b];

n= size(A,1);

x = zeros(n,1);

for i=1:n-1
   for j=i+1:n
      m = A(j,i)/A(i,i)
      A(j,:) = A(j,:) - m*A(i,:)
   end
end
for j=n:-1:2
   for i=j-1:-1:1
      m1=A(i,j)/A(j,j);
      A(i,:)=A(i,:)-m1*A(j,:);
   end
end
//x(1)=A(1,4)/A(1,1)
//x(2)=A(2,4)/A(2,2)
//x(3)=A(3,4)/A(3,3)
for i=1:n
   x(i)=A(i,n+1)/A(i,i);
end
```

Output:


Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console
--> x
x =
1.4
2.4
-0.4
-1.4
-->

Exercise: 7

GAUSS SEIDAL METHOD

Objective:

Find the solution of system of linear equations using Gauss Seidel Method.

Input:

$$a + b + 3c - d = 4$$
$$3b - c + d = 2$$
$$a + 2b - 2c + 5d = 0$$
$$a + b - 5c + 2d = 3$$

Link:

https://images.app.goo.gl/FxnGBUk3qCdoyhpw8

Procedure/Methodology:

$$a + b + 3c - d = 4$$
$$3b - c + d = 2$$
$$a + 2b - 2c + 5d = 0$$
$$a + b - 5c + 2d = 3$$

Converting the equations into matrix form, AX=B

$$\underset{A}{\begin{pmatrix} 1 & 1 & 3 & -1 \\ 0 & 3 & -1 & 4 \\ 1 & 2 & -2 & 5 \\ 1 & 1 & -5 & 2 \end{pmatrix}} \underset{X}{\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}} = \underset{B}{\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}}$$

Augmented Matrix:

$$[A, B] = \left(\begin{array}{cccc|c} 1 & 1 & 3 & -1 & 4 \\ 0 & 3 & -1 & 4 & 2 \\ 1 & 2 & -2 & 5 & 0 \\ 1 & 1 & -5 & 2 & 3 \end{array}\right)$$

The coefficient matrix of the given system is not diagonally dominant.

Hence, we re-arrange the equations as follows, such that the elements in the coefficient matrix are diagonally dominant.

a + b + 3c – d = 4

0a + 3b – c + 4d = 2

a + b - 5c + 2d = 3

a + 2b - 2c + 5d = 0

From the above equations

$a_{k+1} = 1/1[4 - b_k - 3c_k + d_k]$

$b_{k+1} = 1/3[2 - 0a_{k+1} - c_k - 4d_k]$

$c_{k+1} = 1/(-5)[3 - a_{k+10} - b_{k+1} - 2d_k]$

$d_{k+1} = 1/5[0 - a_{k+1} - 2b_{k+1} + 2c_{k+1})]$

Initial gauss (a, b, c, d)=(0,0,0,0)

1st Approximation:

a_1 = 1/1 [4-(0)-3(0)+(0)]	= 1/1[4]	= 4
b_1 = 1/3[2-0(4)+(0)-4(0)]	=1/3[2]	= 0.6667
c_1 = 1/-5[3-(4)-(0.6667)-2(0)]	=1/-5[-1.6667]	= 0.3333
d_1=1/5[0-(4)-2(0.6667)+2(0.3333)]	=1/5[-4.6667]	= -0.9333

2nd Approximation

a_2 = 1/1 [4-(0.6667)-3(0.3333)+(-0.9333)]	=1/1[1.4]	= 1.4
b_2 = 1/3 [2-0(1.4)+(0.3333)-4(-0.9333)]	=1/3[6.0667]	= 2.0222
c_2 = 1/-5 [3-(1.4)-(2.0222)-2(-0.9333)]	=1/-5 [1.4444]	= -0.2889
d_2 = 1/5 [0-(1.4)-2(2.0222)+2(-0.2889)]	=1/5 [-6.0222]	= -1.2044

3rd Approximation

a_3 = 1/1 [4-(2.0222)-3(-0.2889)+(-1.2044)]	= 1/1[1.64]	= 1.64
b_3 = 1/3 [2-0(1.64)+(-0.2889)-4(-1.2044)]	= 1/3[6.5289]	= 2.1763
c_3 = 1/-5 [3-(1.64)-(2.1763)-2(-1.2044)]	= 1/-5[1.5926]	= -0.3185
d_3 = 1/5 [0-(1.64)-2(2.1763)+2(-0.3185)]	= 1/5[-6.6296]	= -1.3259

4th Approximation

a_4 = 1/1 [4-(2.1763)-3(-0.3185)+(-1.3259)]	= 1/1[1.4533]	= 1.4533
b_4 = 1/3 [2-0(1.4533)+(-0.3185)-4(-1.3259)]	= 1/3[6.9852]	= 2.3284
c_4 = 1/-5 [3-(1.4533)-(2.3284)-2(-1.3259)]	= 1/-5[1.8701]	= -0.374
d_4 = 1/5 [0-(1.4533)-2(2.3284)+2(-0.374)]	= 1/5[-6.8582]	= -1.3716

5th Approximation

a_4 = 1/1 [4-(2.3284)-3(-0.374)+(-1.3716)]	= 1/1[1.422]	= 1.422
b_4 = 1/3 [2-0(1.422)+(-0.374)-4(-1.3716)]	= 1/3[7.1125]	= 2.3708
c_4 = 1/-5 [3-(1.422)-(2.3708)-2(-1.3716)]	= 1/-5[1.9504]	= -0.3901
d_4 = 1/5 [0-(1.422)-2(2.3708)+2(-0.3901)]	= 1/5[-6.9439]	= -1.3888

6th Approximation

a_4 = 1/1 [4-(2.3708)-3(-0.3901)+(-1.3888)]	= 1/1[1.4106]	= 1.4106
b_4 = 1/3 [2-0(1.4106)+(-0.3901)-4(-1.3888)]	= 1/3[7.165]	= 2.3883
c_4 = 1/-5 [3-(1.4106)-(2.3883)-2(-1.3888)]	= 1/-5[1.9786]	= -0.3957
d_4 = 1/5 [0-(1.4106)-2(2.3883)+2(-0.3957)]	= 1/5[-6.9787]	= -1.3957

7th Approximation

a_4 = 1/1 [4-(2.3883)-3(-0.3957)+(-1.3957)]	= 1/1[1.4031]	= 1.4031
b_4 = 1/3 [2-0(1.4031)+(-0.3957)-4(-1.3957)]	= 1/3[7.1873]	= 2.3958
c_4 = 1/-5 [3-(1.4031)-(2.3958)-2(-1.3957)]	= 1/-5[1.9927]	= -0.3985
d_4 = 1/5 [0-(1.4031)-2(2.3958)+2(-0.3985)]	= 1/5[-6.9916]	= -1.3983

8th Approximation

a_4 = 1/1 [4-(2.3958)-3(-0.3985)+(-1.3983)]	= 1/1[1.4015]	= 1.4015
b_4 = 1/3 [2-0(1.4015)+(-0.3985)-4(-1.3983)]	= 1/3[7.1948]	= 2.3983
c_4 = 1/-5 [3-(1.4015)-(2.3983)-2(-1.3983)]	= 1/-5[1.9969]	= -0.3994
d_4 = 1/5 [0-(1.4015)-2(2.3983)+2(-0.3994)]	= 1/5[-6.9968]	= -1.3994

9th Approximation

a_4 = 1/1 [4-(2.3983)-3(-0.3994)+(-1.3994)] = 1/1[1.4005] = 1.4005

b_4 = 1/3 [2-0(1.4005)+(-0.3994)-4(-1.3994)] = 1/3[7.1981] = 2.3994

c_4 = 1/-5 [3-(1.4005)-(2.3994)-2(-1.3994)] = 1/-5[1.9989] = -0.3998

d_4 = 1/5 [0-(1.4005)-2(2.3994)+2(-0.3998)] = 1/5[-6.9988] = -1.3998

10th Approximation

a_4 = 1/1 [4-(2.3994)-3(-0.3998)+(-1.3998)] = 1/1[1.4002] = 1.4002

b_4 = 1/3 [2-0(1.4002)+(-0.3998)-4(-1.3998)] = 1/3[7.1992] = 2.3997

c_4 = 1/-5 [3-(1.4002)-(2.3997)-2(-1.3998)] = 1/-5[1.9995] = -0.3999

d_4 = 1/5 [0-(1.4002)-2(2.3997)+2(-0.3999)] = 1/5[-6.9995] = -1.3999

Solution by Gauss Seidel Method.

a = 1.4002 ≅ 1.4

b = 2.3997 ≅ 2.4

c = -0.3999 ≅ -0.4

d = -1.3999 ≅ -1.4

Iterations are tabulated as below

Iteration	**a**	**b**	**c**	**d**
1	4	0.6667	0.3333	-0.9333
2	1.4	2.0222	-0.2889	-1.2044
3	1.64	2.1763	-0.3185	-1.3259
4	1.4533	2.3284	-0.374	-1.3716
5	1.422	2.3708	-0.3901	-1.3888
6	1.4106	2.3883	-0.3957	-1.3957
7	1.4031	2.3958	-0.3985	-1.3983
8	1.4015	2.3983	-0.3994	-1.3994
9	1.4005	2.3994	-0.3998	-1.3998
10	1.4002	2.3997	-0.3999	-1.3999

Result:

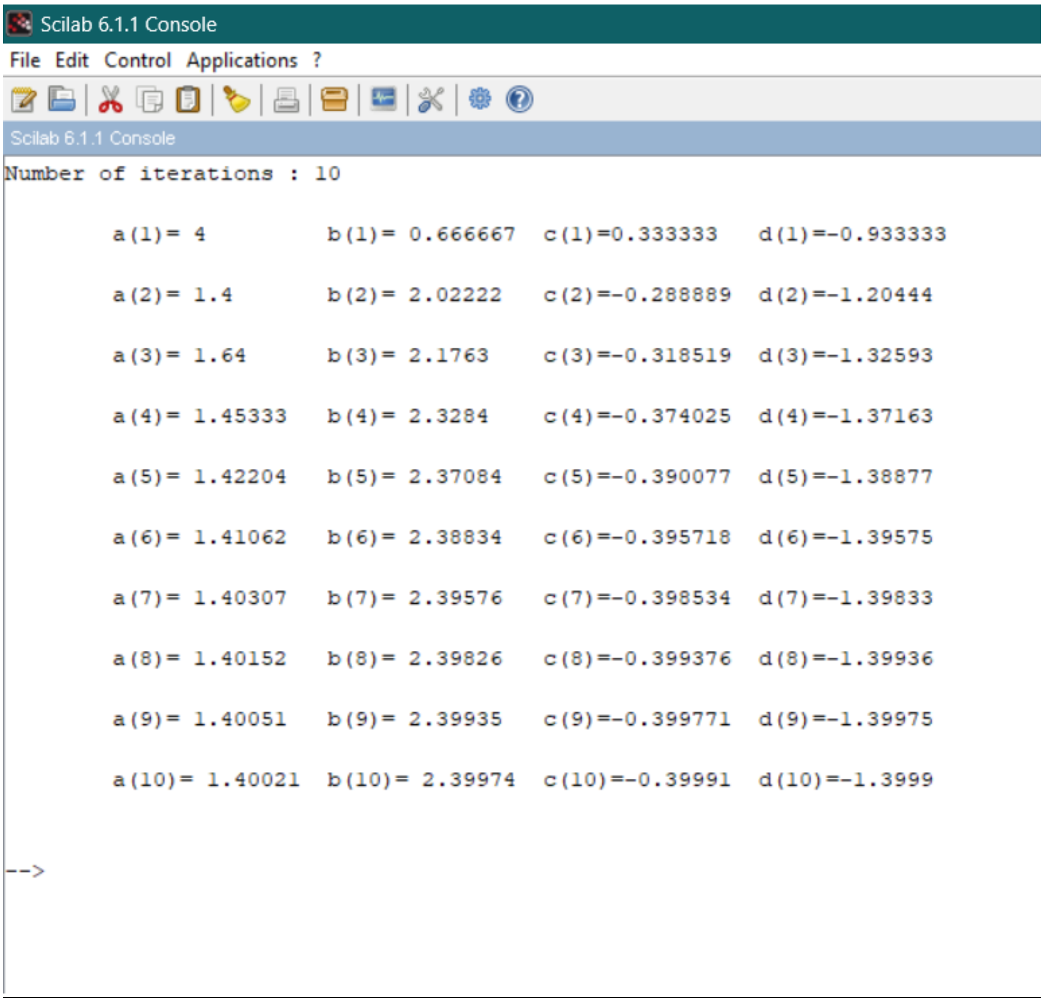
The solution of system of linear equation using Gauss Seidel Method is

a = 1.4; b = 2.4; c = -0.4; d = 1.4

Coding:

```
clc; clear all;
funcprot(0);
deff('a=f1(b,c,d)','a=(4-b-3*c+d)/1')
deff('b=f2(a,c,d)','b=(2+c-4*d)/3')
deff('c=f3(a,b,d)','c=(3-a-b-2*d)/(-5)')
deff('d=f4(a,b,c)','d=(0-a-2*b+2*c)/(5)')
n=10
a0=0;b0=0;c0=0;d0=0;
printf("Number of iterations : %g\n\n",n)
for i=1:n
   a0=f1(b0,c0,d0);
   b0=f2(a0,c0,d0);
   c0=f3(a0,b0,d0);
   d0=f4(a0,b0,c0);
   printf('\ta(%i)= %g \tb(%i)= %g \tc(%i)=%g \td(%i)=%g\n\n',i,a0,i,b0,i,c0,i,d0)
end
```

Output:



MODULE

2

Exercise: 08

NEWTON'S GREGORY FORWARD METHOD

Objective:

To find the solution for the given data by Interpolation using Newton's Gregory Forward Method

Input:

The Database of Milk Production (million tones) in Every Year

Year	Milk Production
1990	54
1991	56
1992	58
1993	61
1994	64
1995	66
1996	69
1997	72
1998	75
1999	78
2000	81

Link:

https://data.gov.in/sites/default/files/datafile/Animal_Husbandary_TABLE1_2012-13.csv

Procedure/Methodology:

Newton’s Forward Difference Table

X	Y	Δy	Δ^2 y	Δ^3 y	Δ^4 y	Δ^5 y	Δ^6 y	Δ^7 y	Δ^8 y	Δ^9 y	Δ^10 y
1990	54										
		2									
1991	56		0								
		2		1							
1992	58		1		-2						
		3		-1		2					
1993	61		0		0		1				
		3		-1		3		-10			
1994	64		-1		3		-9		29		
		2		2		-6		19		-63	
1995	66		1		-3		10		-34		118
		3		-1		4		-15		55	
1996	69		0		1		-5		21		
		3		0		-1		6			
1997	72		0		0		1				
		3		0		0					
1998	75		0		0						
		3		0							
1999	78		0								
		3									
2000	81										

The value of *x* at you want to find the f(x): x =1991.5

$$U= \frac{X-X0}{h}$$

$$= \frac{1991.5-1990}{1} = 1.5$$

Where $h = X_1 - X_0 = 1991-1990 = 1$

Newton’s Forward Difference Interpolation Formula:

$$Y(x) = Yo + \frac{u}{1!}\Delta Yo + \frac{u(u-1)}{2!}\Delta Y^2 o + \frac{u(u-1)(u-2)}{3!}\Delta Y^3 o + \cdots$$

Y (1991.5) = 54+3+0-0.0625-0.0469-0.0234+0.0068+0.0439+0.0876+0.1375+0.1931

$$\boxed{Y(1991.5) = 57.3362}$$

Result:

The solution of Newton’s Gregory Forward Interpolation Method Y(1991.5) = 57.3362

Coding:

```
clc; clear all;
funcprot(0);
x=[1990:2000]
y=[54 56 58 61 64 66 69 72 75 78 81]
xg=input("Enter the x value to find f(x):")
n=length(x)
h=x(2)-x(1)
u=(xg-x(1))/h
disp('Forward difference table:')
disp(x)
disp(y)
for i=1:n-1
   disp(diff(y,i))
end
yg=y(1)
p=u
for i=1:n-1
   d=diff(y,i)
   yg=yg+p*d(1)
   p=p*((u-i)/(i+1))
end
disp('Value of f(1991.5):', yg)
```

Output:

Scilab 6.1.1 Console

File Edit Control Applications ?

Scilab 6.1.1 Console

```
Enter the x value to find f(x):1991.5

  "Forward difference table:"

   1990.   1991.   1992.   1993.   1994.   1995.   1996.   1997.   1998.   1999.   2000.

   54.   56.   58.   61.   64.   66.   69.   72.   75.   78.   81.

   2.   2.   3.   3.   2.   3.   3.   3.   3.   3.

   0.   1.   0.  -1.   1.   0.   0.   0.   0.

   1.  -1.  -1.   2.  -1.   0.   0.   0.

  -2.   0.   3.  -3.   1.   0.   0.

   2.   3.  -6.   4.  -1.   0.

   1.  -9.   10.  -5.   1.

  -10.   19.  -15.   6.

   29.  -34.   21.

  -63.   55.

   118.

  "Value of f(1991.5):"

   57.336159
```

Exercise: 09

NEWTON'S GREGORY BACKWARD METHOD

Objective:

To find the solution for the given data by Interpolation using Newton's Gregory Backward Method

Input:

The Database of Milk Production (million tons) in Every Year

Year	Milk Production
1990	54
1991	56
1992	58
1993	61
1994	64
1995	66
1996	69
1997	72
1998	75
1999	78
2000	81

Link:

https://data.gov.in/sites/default/files/datafile/Animal_Husbandary_TABLE1_2012-13.csv

Procedure/Methodology:

Newton’s Backward Difference Table

X	Y	Δy	Δ^2 y	Δ^3 y	Δ^4 y	Δ^5 y	Δ^6 y	Δ^7 y	Δ^8 y	Δ^9 y	Δ^10 y
1990	54										
		2									
1991	56		0								
		2		1							
1992	58		1		-2						
		3		-1		2					
1993	61		0		0		1				
		3		-1		3		-10			
1994	64		-1		3		-9		29		
		2		2		-6		19		-63	
1995	66		1		-3		10		-34		118
		3		-1		4		-15		55	
1996	69		0		1		-5		21		
		3		0		-1		6			
1997	72		0		0		1				
		3		0		0					
1998	75		0		0						
		3		0							
1999	78		0								
		3									
2000	81										

The value of *x* at you want to find the f(x): x =1998.5

$$U = \frac{X - X0}{h}$$

$$= \frac{1998.5 - 2000}{1} = -1.5$$

Where $h = X_1 - X_0 = 1991 - 1990 = 1$

Newton’s Forward Difference Interpolation Formula:

$$Y(x) = Yo + \frac{u}{1!} \nabla Yo + \frac{u(u+1)}{2!} \nabla Y^2 o + \frac{u(u+1)(u+2)}{3!} \nabla Y^3 o + \cdots$$

Y (1998.5) = 81-4.5+0+0+0+0+0.0068+0.0264+0.0634+0.12+0.1931

$$\boxed{Y(1998.5) = 76.9098}$$

Result:

The solution of Newton’s Gregory Backward Interpolation Method Y (1998.5) = 76.9098

Output:

```
clc; clear all;
funcprot(0);
x=[1990:2000]
y=[54 56 58 61 64 66 69 72 75 78 81]
xg=input("Enter the x value to find f(x):")
n=length(x)
h=x(2)-x(1)
u=(xg-x(n))/h
disp('Backward difference table:')
disp(x)
disp(y)
for i=1:n-1
   disp(diff(y,i))
end
yg=y(n)
p=u
for i=1:n-1
   d=diff(y,i)
   yg=yg+p*d(n-i)
   p=p*((u+i)/(i+1))
end
disp('Value of f(x):', yg)
```

Coding:

```
Scilab 6.1.1 Console
File  Edit  Control  Applications  ?
Scilab 6.1.1 Console

Enter the x value to find f(x):1998.5

  "Backward difference table:"

   1990.   1991.   1992.   1993.   1994.   1995.   1996.   1997.   1998.   1999.   2000.

   54.   56.   58.   61.   64.   66.   69.   72.   75.   78.   81.

   2.   2.   3.   3.   2.   3.   3.   3.   3.   3.

   0.   1.   0.  -1.   1.   0.   0.   0.   0.

   1.  -1.  -1.   2.  -1.   0.   0.   0.

  -2.   0.   3.  -3.   1.   0.   0.

   2.   3.  -6.   4.  -1.   0.

   1.  -9.   10.  -5.   1.

  -10.   19.  -15.   6.

   29.  -34.   21.

  -63.   55.

   118.

  "Value of f(x):"

   76.909767

-->
```

Exercise: 10

NEWTON'S DIVIDED DIFFERENCE METHOD

Objective:

To find the solution for the given data by Interpolation using Newton's Divided Difference Method

Input:

The Database of Milk Production (million tons) in Every Year

Year	Milk Production
1990	54
1991	56
1993	61
1994	64
1997	72
1999	78
2000	81
2002	86
2003	88
2006	103
2009	116

Link:

https://data.gov.in/sites/default/files/datafile/Animal_Husbandary_TABLE1_2012-13.csv

Procedure/Methodology:

Newton’s Divided Difference Table

X	Y	1st order	2nd order	3rd order	4th order	5th order	6th order	7th order
1990	54							
		2						
1991	56		0.1667					
		2.5		0				
1993	61		0.1667		-0.006			
		3		-0.0417		0.0016		
1994	64		-0.0833		0.0083		-0.0003	
		2.6667		0.025		-0.0015		0
1997	72		0.0667		-0.0052		0.0002	
		3		-0.0111		0.0003		0
1999	78		0		-0.0028		0	
		3		-0.0333		0.0009		0
2000	81		-0.1667		0.0056		0	
		2.5		0		0.0018		0
2002	86		-0.1667		0.0218		-0.0006	
		2		0.1528		-0.0052		
2003	88		0.75		-0.0306			
		5		-0.123				
2006	103		-0.1111					
		4.3333						
2009	116							

The value of *x* at you want to find the f(x):x=1995

Newton's divided difference interpolation formula is

f(x) =y0+(x-x0) f[x0,x1]+(x-x0)(x-x1)f[x0,x1,x2]+(x-x0)(x-x1)(x-x2)f[x0,x1,x2,x3]+...

Y (1995) = 54+10+3.334+0-0.24-0.128-0.096+0

$\boxed{\text{Y (1995) = 66.87}}$

Result:

The solution of Newton’s Divided Difference Interpolation Method Y (1995) = 66.87

Coding:

```
clc; clear all;
n = input('Enter n for n+1 nodes,n:');
x = zeros(1,n+1);
y = zeros(n+1,n+1);
for i = 0:n
printf('Enter x(%d) and f(x(%d)) on separate lines: \n', i, i);
x(i+1) = input(' ');
y(i+1,1) = input(' ');
end
x0 = input('Now enter a point at which to evaluate the polynomial, x = ')
n = size(x,1);
if(n==1)
n = size(x,2);
end
for i = 1:n
D(i,1) = y(i);
end
for i = 2:n
for j = 2:i
D(i,j)=(D(i,j-1)-D(i-1,j-1))/(x(i)-x(i-j+1));
end
end
fx0 = D(n,n);
for i = n-1:-1:1
fx0 = fx0*(x0-x(i)) + D(i,i);
end
printf('Newtons iterated value: %.4f \n', fx0)
```

<u>Output:</u>

Scilab 6.1.1 Console

File Edit Control Applications ?

Scilab 6.1.1 Console

```
Enter n for n+1 nodes,n:10

Enter x(0) and f(x(0)) on separate lines:
 1990

 54

Enter x(1) and f(x(1)) on separate lines:
 1991

 56

Enter x(2) and f(x(2)) on separate lines:
 1993

 61

Enter x(3) and f(x(3)) on separate lines:
 1994

 64

Enter x(4) and f(x(4)) on separate lines:
 1997

 72

Enter x(5) and f(x(5)) on separate lines:
 1999

 78

Enter x(6) and f(x(6)) on separate lines:
 2000

 81

Enter x(7) and f(x(7)) on separate lines:
 2002

 86

Enter x(8) and f(x(8)) on separate lines:
 2003

 88

Enter x(9) and f(x(9)) on separate lines:
 2006

 103

Enter x(10) and f(x(10)) on separate lines:
 2009

 116

Now enter a point at which to evaluate the polynomial, x = 1995

Newtons iterated value: 66.7200
```

Exercise: 11

LAGRANGE'S INTERPOLATION METHOD

Objective:

To find the solution for the given data by Lagrange's Interpolation Method

Input:

The Database of Milk Production (million tons) in Every Year

Year	Milk Production
1990	54
1991	56
1993	61
1994	64
1997	72
1999	78
2000	81
2002	86
2003	88
2006	103
2009	116

Link:

https://data.gov.in/sites/default/files/datafile/Animal_Husbandary_TABLE1_2012-13.csv

Procedure/Methodology:

The value of table for x and y

x	1990	1991	1993	1994	1997	1999	2000	2002	2003	2006	2009
y	54	56	61	64	72	78	81	86	88	103	116

The value of x at you want to find f(x): x =1995

Lagrange's Interpolation formula is f(x)

$$\frac{(x-x_1)(x-x_2)(x-x_3)\ldots.(x-x_{11})}{(x_0-x_1)(x_0-x_2)(x_0-x_3)\ldots.(x_0-x_{11})}+\frac{(x-x_0)(x-x_2)(x-x_3)\ldots.(x-x_{11})}{(x_1-x_1)(x_1-x_2)(x_1-x_3)\ldots.(x_1-x_{11})}\ldots..$$

$$+\frac{(x-x_1)(x-x_2)(x-x_3)\ldots.(x-x_{10})}{(x_{11}-x_1)(x_{11}-x_2)(x_{11}-x_3)\ldots.(x_{11}-x_{10})}$$

Y (1995) = (-0.0077) ×54+0.0373×56+(-0.3656) ×61+0.9858×64+0.7042×72+(-0.9506) × 78+ 0.7511×81+(-0.2469)×86+0.0948×88+(-0.0025)×103+0.0001×116

Y (1995)=66.72

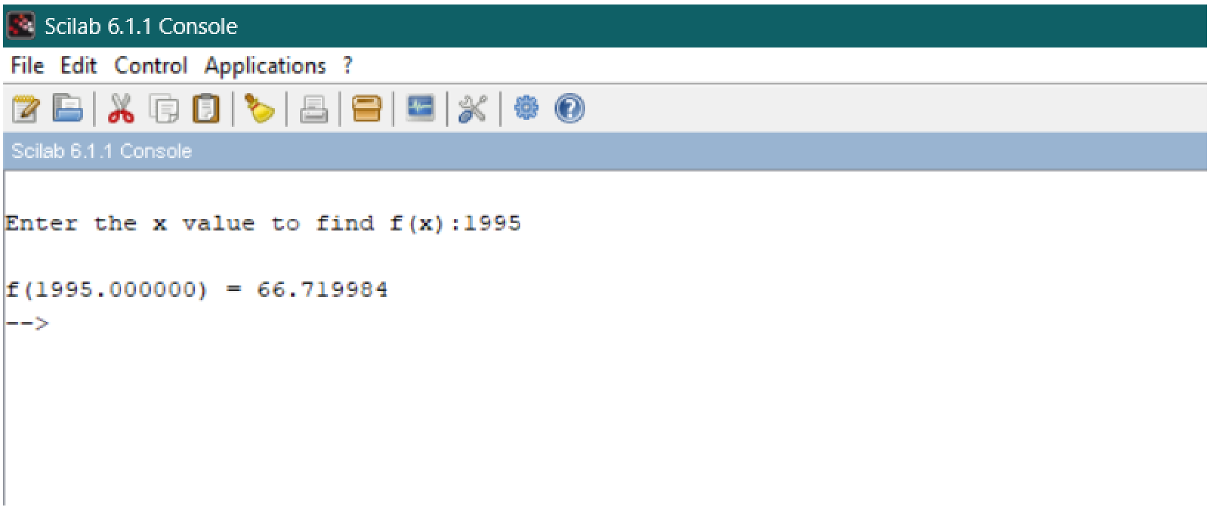
Result:

The solution of Lagrange's Interpolation Method Y (1995) = 66.72

Output:

```
clc; clear all;
funcprot(0);
X=[1990 1991 1993 1994 1997 1999 2000 2002 2003 2006 2009]
Y=[54 56 61 64 72 78 81 86 88 103 116]
x=input("Enter the x value to find f(x):")
n=length(X);
L=0;
for i=1:n
   N=1;D=1;
   for j=1:n
      if(i==j)
         continue;
      else
         N=N*(x-X(j));
         D=D*(X(i)-X(j));
      end
   end
   L=L+(N/D)*Y(i);
end
printf("f(%f) = %f",x,L);
```

Output:


Scilab 6.1.1 Console
File Edit Control Applications ?
Scilab 6.1.1 Console
Enter the x value to find f(x):1995
f(1995.000000) = 66.719984
-->

MODULE

3

Exercise: 12

NEWTON'S FORWARD METHOD

Objective:

To find the solution for the given data by Interpolation using Newton's Forward Method

Input:

The Database of Milk Production (million tones) in Every Year

Year	Milk Production
1990	54
1991	56
1992	58
1993	61
1994	64
1995	66
1996	69
1997	72
1998	75
1999	78
2000	81

Link:

https://data.gov.in/sites/default/files/datafile/Animal_Husbandary_TABLE1_2012-13.csv

Procedure/Methodology:

Newton’s Forward Difference Table

X	Y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$	$\Delta^6 y$	$\Delta^7 y$	$\Delta^8 y$	$\Delta^9 y$	$\Delta^{10} y$
1990	54										
		2									
1991	56		0								
		2		1							
1992	58		1		-2						
		3		-1		2					
1993	61		0		0		1				
		3		-1		3		-10			
1994	64		-1		3		-9		29		
		2		2		-6		19		-63	
1995	66		1		-3		10		-34		118
		3		-1		4		-15		55	
1996	69		0		1		-5		21		
		3		0		-1		6			
1997	72		0		0		1				
		3		0		0					
1998	75		0		0						
		3		0							
1999	78		0								
		3									
2000	81										

The value of *x* at you want to find the f(x): x =1991.5

$$t = \frac{X - X_0}{h}$$

$$= \frac{1991.5 - 1990}{1} = 1.5$$

Where $h = X_1 - X_0 = 1991 - 1990 = 1$

Newton’s Forward Difference Interpolation Formula:

$$\left(\frac{dy}{dx}\right)_{X=Xo} = \frac{1}{h}\left(\Delta Yo + \frac{2t-1}{2!}\Delta^2 Yo + \frac{3t^2-6t+2}{3!}\Delta^3 Yo + \frac{4t^3-18t^2+22t-6}{4!}\Delta^4 Yo\right)$$

$$\left(\frac{dy}{dx}\right)_{X=1991.5} = \frac{1}{1}\left(2 + \frac{2}{2} \text{ x } 0 + \frac{-0.25}{6} \text{ x } 1 + \frac{0}{24} \text{ x } -2\right)$$

$$\boxed{\left(\frac{dy}{dx}\right)_{X=1991.5} = 1.9583}$$

$$\left(\frac{d^2y}{dx^2}\right)_{X=Xo} = \frac{1}{h^2}\ \Delta^2 Yo + (t-1)\Delta^3 Yo + \frac{12t^2-36t+22}{24}\Delta^4 Yo$$

$$\left(\frac{d^2y}{dx^2}\right)_{X=1991.5} = \frac{1}{1}\left(0 + 0.5 \text{ x } 1 + \frac{5}{24} \text{ x} - 2\right)$$

$$\boxed{\left(\frac{d^2y}{dx^2}\right)_{X=1991.5} = 0.9167}$$

Result:

The solution of Newton's Forward Method is f ′(1991.5) , Velocity = 1.9583 and

f ″(1991.5), Acceleration = 0.9167

Coding:

```
clc;
clear all;
x=[1990:2000];
y=[54,56,58,61,64,66,69,72,75,78,81];
n=length(x);
xg=1991.5;
h=x(2)-x(1);
p=(xg-x(1)/h);
d=%nan*ones(n,6);
d(:,1)=y';
for j=2:6
    for i=1:n-j+1
        d(i,j)=d(i+1,j-1)- d(i,j-1);
    end
end
mprintf("%5s %6s %9s %8s %8s %7s",'x','y','dy','d2y','d3y','d4y','d5y')
disp([x',d])
dy=(1/h)*[(d(1,2)+((2*p-1)/2)*d(1,3)+((3*p^2-6*p+2)/6)*d(1,4)+((4*p^3-18*p^2+22*p-
6)/24)*d(1,5))]
d2y=(1/h^2)*[(d(1,3)+(p-1)*d(1,4)+(12*p^2-36*p+22)/24*d(1,5))]
printf("\n Velocity
 is % 4f \n\n",dy);
printf("\n Acceleration is %4f \n\n",d2y);
```

Output:

Scilab 6.1.1 Console

File Edit Control Applications ?

Scilab 6.1.1 Console

```
    x         y        dy        d2y       d3y       d4y
   1990.     54.    2.     0.     1.    -2.     2.
   1991.     56.    2.     1.    -1.     0.     3.
   1992.     58.    3.     0.    -1.     3.    -6.
   1993.     61.    3.    -1.     2.    -3.     4.
   1994.     64.    2.     1.    -1.     1.    -1.
   1995.     66.    3.     0.     0.     0.     0.
   1996.     69.    3.     0.     0.     0.     Nan
   1997.     72.    3.     0.     0.     Nan    Nan
   1998.     75.    3.     0.     Nan    Nan    Nan
   1999.     78.    3.     Nan    Nan    Nan    Nan
   2000.     81.    Nan    Nan    Nan    Nan    Nan

 velocity is  1.958333


 acceleration id 0.916667
```

Exercise: 13

NEWTON'S BACKWARD METHOD

Objective:

To find the solution for the given data by Interpolation using Newton's Backward Method

Input:

The Database of Milk Production (million tons) in Every Year

Year	Milk Production
1990	54
1991	56
1992	58
1993	61
1994	64
1995	66
1996	69
1997	72
1998	75
1999	78
2000	81

Link:

https://data.gov.in/sites/default/files/datafile/Animal_Husbandary_TABLE1_2012-13.csv

Procedure/Methodology:

Newton’s Backward Difference Table

X	Y	Δy	Δ^2 y	Δ^3 y	Δ^4 y	Δ^5 y	Δ^6 y	Δ^7 y	Δ^8 y	Δ^9 y	Δ^10 y
1990	54										
		2									
1991	56		0								
		2		1							
1992	58		1		-2						
		3		-1		2					
1993	61		0		0		1				
		3		-1		3		-10			
1994	64		-1		3		-9		29		
		2		2		-6		19		-63	
1995	66		1		-3		10		-34		118
		3		-1		4		-15		55	
1996	69		0		1		-5		21		
		3		0		-1		6			
1997	72		0		0		1				
		3		0		0					
1998	75		0		0						
		3		0							
1999	78		0								
		3									
2000	81										

The value of *x* at you want to find the f(x): x =1998.5

$$t = \frac{X - X0}{h}$$

$$= \frac{1998.5 - 2000}{1} = -1.5$$

Where $h = X_1 - X_0$ = 1991-1990 = 1

Newton’s Backward Difference Interpolation Formula:

$$\left(\frac{dy}{dx}\right)_{X=Xn} = \frac{1}{h}\left(\nabla Yn + \frac{2t+1}{2!}\nabla^2 Yo + \frac{3t^2+6t+2}{3!}\nabla^3 Yo + \frac{4t^3+18t^2+22t+6}{4!}\nabla^4 Yo\right)$$

$$\left(\frac{dy}{dx}\right)_{X=1998.5} = \frac{1}{1}\left(3 + \frac{-2}{2}\text{ x } 0 + \frac{-0.25}{6}\text{ x } 1 + \frac{0}{24}\text{ x } 0\right)$$

$$\boxed{\left(\frac{dy}{dx}\right)_{X=1998.5} = 3}$$

$$\left(\frac{d^2y}{dx^2}\right)_{X=Xn} = \frac{1}{h^2}\left(\nabla^2 Yn + (t+1)\nabla^3 Yo + \frac{12t^2+36t+22}{24}\nabla^4 Yo\right)$$

$$\left(\frac{d^2y}{dx^2}\right)_{X=1998.5} = \frac{1}{1}\left(0 + (-0.5)\text{x } 0 + \frac{-5}{24}\text{ x}0\right)$$

$$\boxed{\left(\frac{d^2y}{dx^2}\right)_{X=1998.5} = 0}$$

Result:

The solution of Newton's Forward Method is f '(1998.5), Velocity = 3 and f "(1991.5), Acceleration =0

Coding:

```
clc;
clear all;
x=[1990:2000];
y=[54,56,58,61,64,66,69,72,75,78,81];
n=length(x);
xg=1998.5;
h=x(2)-x(1);
p=(xg-x(n))/h;
d=%nan*ones(n,6);
d(:,1)=y';
for j=2:6
   for i=1:n-j+1
      d(i,j)=d(i+1,j-1)- d(i,j-1);
   end
end
mprintf("%5s %6s %9s %8s %8s %8s %7s",'x','y','dy','d2y','d3y','d4y','d5y')
disp([x',d])
dy=(1/h)*[(d(19)+((2*p+1)/2)*d(28)+((3*p^2+6*p+2)/6)*d(37)+((4*p^3+18*p^2+22*p+6)/
24)*d(46))]
d2y=(1/h^2)*[(d(28)+(p-1)*d(39)+(12*p^2+36*p+22)/24*d(46))]
printf("\n Velocity is % 4f \n\n",dy);
printf("\n Acceleration is %4f \n\n",d2y);
```

Output:

Scilab 6.1.1 Console

File Edit Control Applications ?

Scilab 6.1.1 Console

```
    x        y          dy        d2y       d3y       d4y      d5y
   1990.    54.    2.     0.     1.    -2.     2.
   1991.    56.    2.     1.    -1.     0.     3.
   1992.    58.    3.     0.    -1.     3.    -6.
   1993.    61.    3.    -1.     2.    -3.     4.
   1994.    64.    2.     1.    -1.     1.    -1.
   1995.    66.    3.     0.     0.     0.     0.
   1996.    69.    3.     0.     0.     0.     Nan
   1997.    72.    3.     0.     0.     Nan    Nan
   1998.    75.    3.     0.     Nan    Nan    Nan
   1999.    78.    3.     Nan    Nan    Nan    Nan
   2000.    81.    Nan    Nan    Nan    Nan    Nan

 Velocity is  2.916667


 Acceleration is 0.000000
```

Exercise:14

TRAPEZOIDAL METHOD

Objective:

To integrate the given function using Trapezoidal Method.

Input:

$$\int_{-2}^{5} \sqrt{x^2+1}\, dx$$

Link:

https://vivadifferences.com/difference-between-trapezoidal-rule-and-simpsons-rule-in-surveying/

Procedure/Methodology:

$$f(x) = \sqrt{x^2+1}$$

Where, a = -2, b = 5 and n = 6

$$h = \frac{b-a}{n} = 1.1667$$

The value of table for *x* and *y*

x	-2	-0.8333	0.3333	1.5	2.6667	3.8333	5
y	2.2361	1.3017	1.0541	1.8028	2.848	3.9616	5.099

Using Trapezoidal Rule,

$\int$ y dx = h /2 [y0 + y6 + 2 (y1 + y2 + y3 + y4 + y5)]

$\int$ y dx=1.1667 / 2 [2.2361 + 5.099 + 2 × (1.3017+1.0541+1.8028+2.848+3.9616)]

$\int$y dx=1.1667 / 2 [2.2361 +5.099 + 2 × (10.9682)]

$\boxed{\int y\, dx = 17.075}$

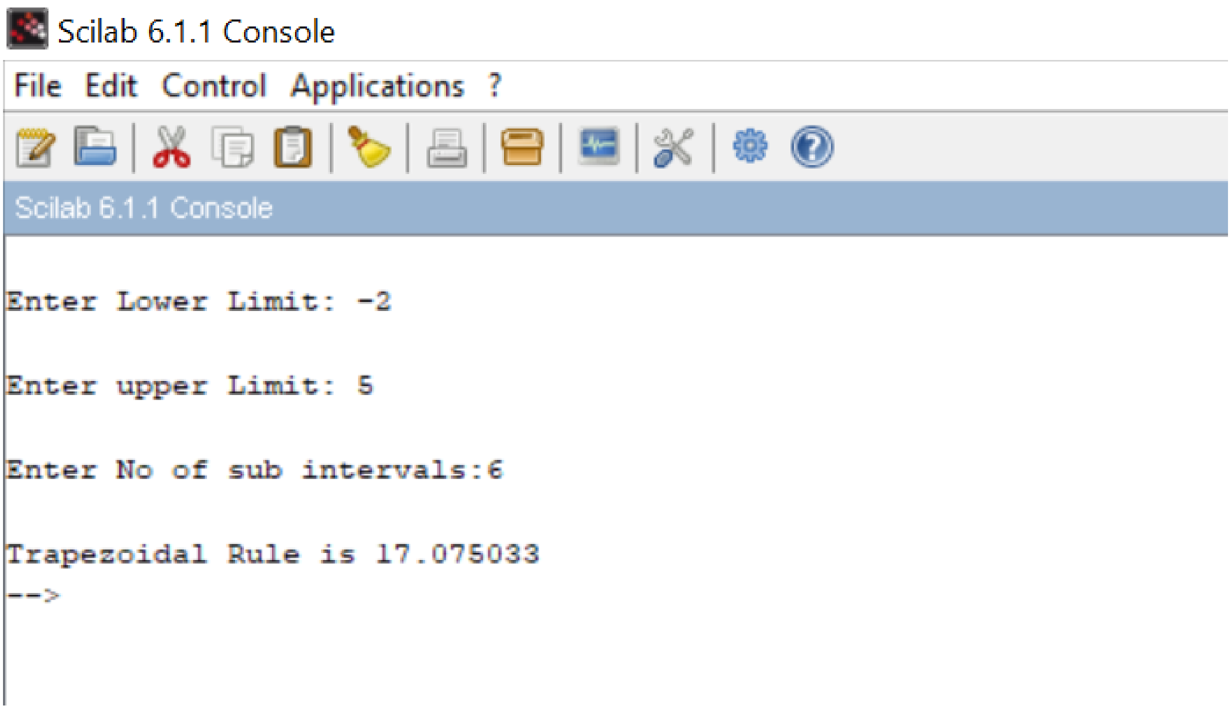
Result:

The Solution for the given function by integration using Trapezoidal Rule is 17.075

Coding:

```
clc; clear all;
deff("y=f(x)","y=(x^2+1)^(1/2)")
a=input("Enter Lower Limit: ")
b=input("Enter upper Limit: ")
n=input("Enter No of sub intervals:")
h=(b-a)/n;
sum1=0;
for i=1:n-1
    x=a+i*h;
    sum1=sum1+f(x);
end
I=(h/2)*(f(a)+f(b)+2*sum1);
//disp(I);
printf("Trapezoidal Rule is %f",I)
```

Output:



Exercise: 15

SIMPSON'S $^{1}/_{3}\ ^{th}$METHOD

Objective:

To integrate the given function using Simpon's $^{1}/_{3}$ th Method.

Input:

$\int_{-2}^{5} \sqrt{x^2 + 1}$ dx

Link:

https://vivadifferences.com/difference-between-trapezoidal-rule-and-simpsons-rule-in-surveying/

Procedure/Methodology:

f(x) = $\sqrt{x^2 + 1}$

Where, a = -2 , b = 5 and n = 6

h = $\frac{b-a}{n}$ = 1.1667

The value of table for *x* and *y*

x	-2	-0.8333	0.3333	1.5	2.6667	3.8333	5
y	2.2361	1.3017	1.0541	1.8028	2.848	3.9616	5.099

Using Simpsons $^{1}/_{3}$ th Rule

∫ y dx = h / 3[(y0 + y6) + 4 (y1 + y3 + y5) + 2 (y2 + y4)]

∫ y dx =1.1667 / 3[(2.2361+5.099)+4×(1.3017+1.8028+3.9616)+2×(1.0541+2.848)]

∫ y dx = 1.1667 / 3[(2.2361+5.099)+4×(7.0661)+2×(3.9021)]

∫ y dx = 16.8792

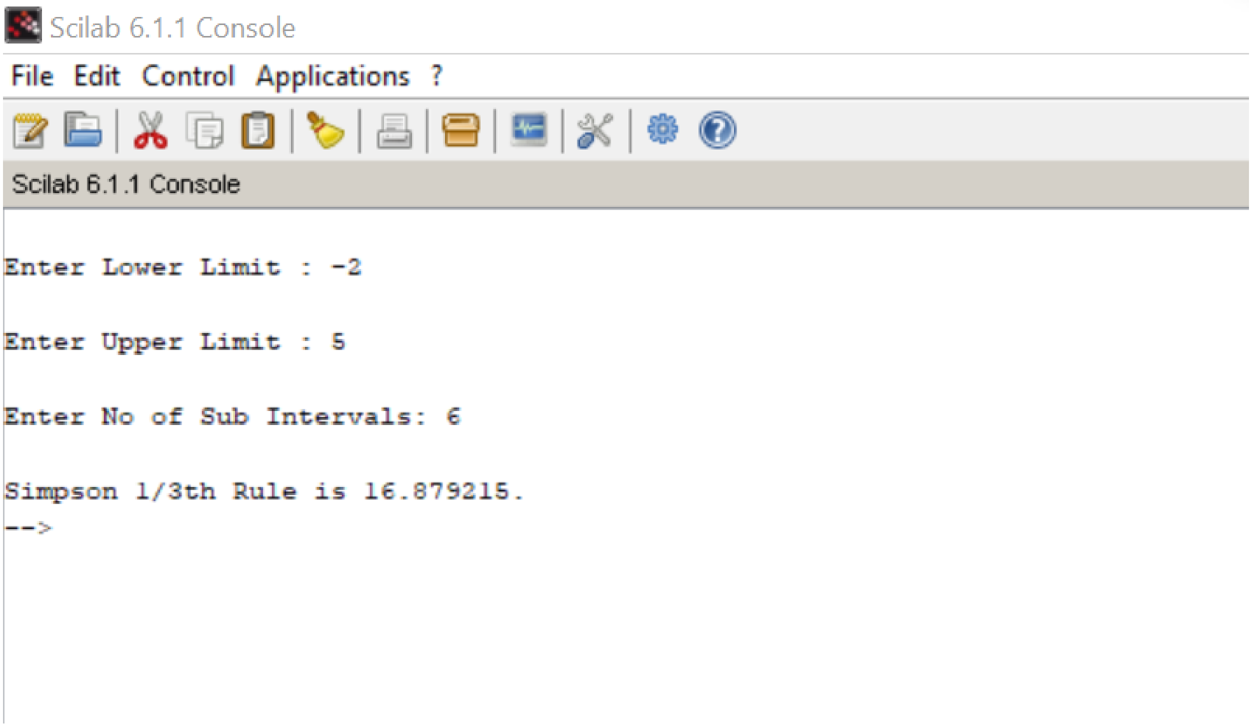
Result:

The Solution for the given function by integration using Simpon's $^{1}/_{3}$ th Rule is 16.8792

Coding:

```
clc;clear all;
deff("y=f(x)","y=(x^2+1)^(1/2)")
a=input("Enter Lower Limit : ")
b=input("Enter Upper Limit : ")
n=input("Enter No of Sub Intervals: ")
h=(b-a)/n;
sum1=0;
for i=1:n-1
    x=a+i*h;
    if modulo(i,2)==0
        sum1=sum1+2*f(x);
    else
        sum1=sum1+4*f(x)
    end
end
I=(h/3)*(f(a)+f(b)+sum1);
//disp(I);
printf("Simpson 1/3th Rule is %f",I)
```

Output:



Exercise: 16

SIMPSON'S $^{3}/_{8}$ thMETHOD

Objective:

To integrate the given function using Simpon's $^{3}/_{8}$ th Method.

Input:

$$\int_{-2}^{5} \sqrt{x^2 + 1}\, dx$$

Link:

https://vivadifferences.com/difference-between-trapezoidal-rule-and-simpsons-rule-in-surveying/

Procedure/Methodology:

$f(x) = \sqrt{x^2 + 1}$

Where, a = -2 , b = 5 and n = 6

$h = \frac{b-a}{n} = 1.1667$

The value of table for *x* and *y*

x	-2	-0.8333	0.3333	1.5	2.6667	3.8333	5
y	2.2361	1.3017	1.0541	1.8028	2.848	3.9616	5.099

Using Simpsons $^{3}/_{8}$ th Rule

∫ y dx = 3h / 8 [(y0 + y6) + 2 (y3) + 3 (y1 + y2 + y4 + y5)]

∫ y dx = 3×1.1667 / 8[(2.2361+5.099)+2×(1.8028)+3×(1.3017+1.0541+2.848+3.9616)]

∫ y dx = 3×1.1667 / 8[(2.2361+5.099)+2×(1.8028)+3×(9.1654)]

$\boxed{\int y\, dx = 16.8161}$

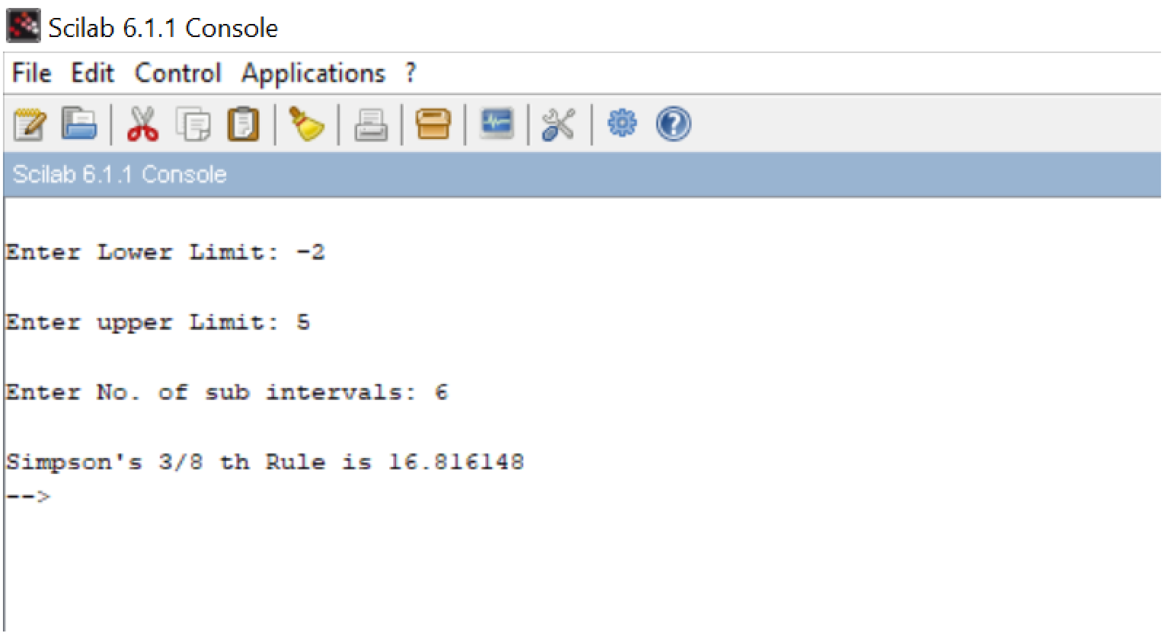
Result:

The Solution for the given function by integration using Simpon's $^{3}/_{8}$ th Rule is 16.8161

Coding:

```
clc; clear all;
deff("y=f(x)",'y=(x^2+1)^(1/2)')
a=input("Enter Lower Limit: ")
b=input("Enter upper Limit: ")
n=input("Enter No. of sub intervals: ")
h=(b-a)/n;
sum1=0;
for i=1:n-1
    x=a+i*h;
    if modulo(i,3)==0
        sum1=sum1+2*f(x);
    else
        sum1=sum1+3*f(x)
        end
end
I=(3*h/8)*(f(a)+f(b)+sum1);
//disp(I);
printf("Simpson"s 3/8 th Rule is %f",I)
```

Output:



MODULE

4

Exercise: 17

TAYLOR’S METHOD

Objective:

To find the solution for the given equation using Taylor Series Method.

Input:

$$y' = x^3 + 9x^2 - 10x + 2$$

Where y(x) = 0, with step length =1. To find Xn = 4

Link:

https://tutorial.math.lamar.edu/ProblemsNS/CalcII/TaylorSeries.aspx

Procedure/Methodology:

Given , $y' = x^3 + 9x^2 - 10x + 2$;

y (0) =0, h=1, y (3) =?

Here $x_0 = 0$; $y_0 = 0$; h = 1 ; $x_n = 3$

Taylor’s Series Formula:

$$y_n = y_{n-1} + hy'_{n-1} + \frac{h^2}{2!}y''_{n-1} + \frac{h^3}{3!}y'''_{n-1} + \cdots$$

Differentiating successively, we get Derivative steps

$$Y' = x^3 + 9x^2 - 10x + 2$$

$$Y'' = 3x^2 + 18x - 10$$

$$Y''' = 6x + 18$$

$$Y'''' = 6$$

Taking (X_0, Y_0)

$Y_0' = {x_0}^3 + 9{x_0}^2 - 10x_0 + 2$ = 2

$Y_0'' = 3{x_0}^2 + 18x_0 - 10$ = -10

$Y_0''' = 6x_0 + 18$ = 18

$Y_0'''' = 6$ = 6

Putting the Values in Taylor series,

$$y_1 = y_0 + hy_0' + \frac{h^2}{2!}y_0'' + \frac{h^3}{3!}y_0''' + \frac{h^4}{4!}y_0''''$$

= 0 + 1(2) + 0.5(-10) + (-0.1667) (18) + (0.0417)(6)

= 0 +2 – 5 + 3 – 0.25

= 0.25

$\boxed{Y(1) = 0.25}$

Taking (X_1, Y_1)

$Y_1' = x_1^3 + 9x_1^2 - 10x_1 + 2$ = 2

$Y_1'' = 3x_1^2 + 18x_1 - 10$ = 11

$Y_1''' = 6x_1 + 18$ = 24

$Y_1'''' = 6$ = 6

Putting the Values in Taylor series,

$$y_2 = y_1 + hy_1' + \frac{h^2}{2!}y_1'' + \frac{h^3}{3!}y_1''' + \frac{h^4}{4!}y_1''''$$

= 0.25 + 1(2) + 0.5(11) + (-0.1667) (24) + (0.0417)(6)

= 0.25 +2 + 5.5 + 4 + 0.25

= 12

$\boxed{Y(2) = 12}$

Taking (X_2, Y_2)

$Y_2' = x_2^3 + 9x_2^2 - 10x_2 + 2$ = 26

$Y_2'' = 3x_2^2 + 18x_2 - 10$ = 38

$Y_2''' = 6x_2 + 18$ = 30

$Y_2'''' = 6$ = 6

Putting the Values in Taylor series,

$$y_3 = y_2 + hy_2' + \frac{h^2}{2!}y_2'' + \frac{h^3}{3!}y_2''' + \frac{h^4}{4!}y_2''''$$

= 12 + 1(26) + 0.5(38) + (-0.1667) (30) + (0.0417)(6)

= 12 + 26 + 19 + 5 +0.25

= 12

$\boxed{Y(2) = 62.25}$

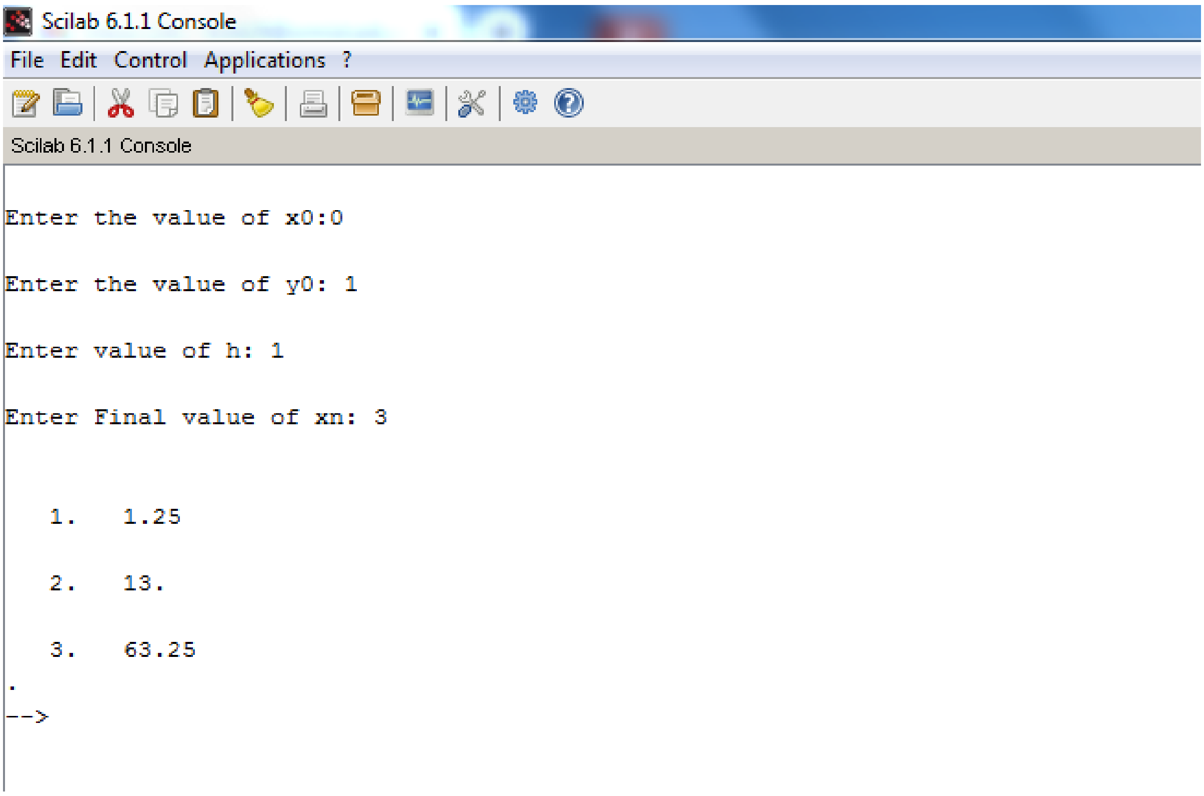
Result:

The Solution of Y(3) is 62.25 by using Taylor Series Method

Coding:

```
clc; clear all;
deff("z=f(x,y)","z=x^3+9*x^2-10*x+2")
deff("z1=f1(x,y)","z1=3*x^2+18*x-10")
deff("z2=f2(x,y)","z2=6*x+18")
deff("z3=f3(x,y)","z3=6")
x0=input("Enter the value of x0:")
y0=input("Enter the value of y0: ")
h=input("Enter value of h: ")
xn=input("Enter Final value of xn: ")
N=(xn-x0)/h
for i=1:N
y1=y0+h*f(x0,y0)+((h^2)/2)*f1(x0,y0)+((h^3)/6)*f2(x0,y0)+((h^4)/24)*f3(x0,y0)
x0=x0+h
disp([x0 y1])
y0=y1
end
halt(".")
```

Output:



Exercise: 18

EULER METHOD

Objective:

To find the solution for the given equation using Euler Method.

Input:

$$y' = x^3 + 9x^2 - 10x + 2$$

Where y(x) = 0, with step length =1. To find Xn = 4

Link:

https://tutorial.math.lamar.edu/ProblemsNS/CalcII/TaylorSeries.aspx

Procedure/Methodology:

Given , $y' = x^3 + 9x^2 - 10x + 2$;

y (0) =0, h=1, y (3) =?

Here $x_0 = 0$; $y_0 = 0$; h = 1 ; $x_n = 3$

Euler Formula:

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

Taking (X_0, Y_0)

$$y_1 = y_0 + \mathrm{h\ f}\,(x_0, y_0)$$

= 0 + 1 f (0, 0)

= 0 + 1(2)

$$\boxed{y_1 = 2}$$

Taking (X_1, Y_1)

$$y_2 = y_1 + \mathrm{h\ f}\,(x_1, y_1)$$

= 2 + 1 f (1, 2)

= 2 + 1(2)

$$\boxed{y_2 = 4}$$

Taking (X_2, Y_2)

$$y_3 = y_2 + h\, f\,(x_2, y_2)$$

$$= 4 + 1\, f\,(2, 4)$$

$$= 4 + 1(26)$$

$$\boxed{y_3 = 30}$$

Result:

The Solution of Y(3) is 30 by using Euler Method

Coding:

```
clc; clear all;
function ydot=f(x, y)
   ydot=x^3+9*x^2-10*x+2
endfunction
x0=input("Enter initial value of x0: ")
y0=input("Enter the value of y0: ")
h=input("Enter value of h: ")
xn=input("Enter Final value of xn: ")
N=(xn-x0)/h
function [x, y]=euler(f, x0, y0, h, N)
   x=zeros(N+1,1)
   y=zeros(N+1,1)
   x(1)=x0
   y(1)=y0
   for j=1:N
      x(j+1)=x(j)+h
      y(j+1)=y(j)+h*f(x(j),y(j))
   end
endfunction
```

Output:

```
Scilab 6.1.1 Console
File  Edit  Control  Applications  ?
Scilab 6.1.1 Console

Enter initial value of x0: 0

Enter the value of y0: 0

Enter value of h: 1

Enter Final value of xn: 3


--> [x,y]=euler(f, x0, y0, h, N)
 x  =

   0.
   1.
   2.
   3.
 y  =

   0.
   2.
   4.
   30.

-->
```

Exercise: 18

IMPROVED EULER METHOD

Objective:

To find the solution for the given equation using Improved Euler Method.

Input:

$$y' = x^3 + 9x^2 - 10x + 2$$

Where y(x) = 0, with step length =1. To find Xn = 4

Link:

https://tutorial.math.lamar.edu/ProblemsNS/CalcII/TaylorSeries.aspx

Procedure/Methodology:

Given , $y' = x^3 + 9x^2 - 10x + 2$;

y (0) =0, h=1, y (3) =?

Here $x_0 = 0$; $y_0 = 0$; h = 1 ; $x_n = 3$

Improved Euler Formula:

$$y_{n+1} = y_n + \frac{1}{2} h \, [\, f \, (x_n , y_n) + f \, (x_n + h \, , y_n + h(x_n , y_n)$$

Taking (X_0, Y_0)

f (x_0 , y_0)

= f (0,0) =2

f $(x_0 + h \, , y_0 + h(x_0 , y_0)$

= f (1,2) =2

$$y_1 = y_0 + \frac{1}{2} h \, [\, f \, (x_0 , y_0) + f \, (x_0 + h \, , y_0 + h(x_0 , y_0)$$

= 0 + 0.5 [2 + f (1,2)]

= 0.5 [2 + 2]

$\boxed{y_1 = 2}$

Taking (X_1, Y_1)

f (x_1 , y_1)

= f (1,2) = 2

f $(x_1 + h\, , y_1 + h(x_1\, , y_1)$

= f (2, 4) = 26

$y_2 = y_1 + \frac{1}{2}$h [f $(x_1\, , y_1)$ + f $(x_1 + h\, , y_1 + h(x_1\, , y_1)$

= 2 + 0.5 [2 + 26]

$\boxed{y_2 = 16}$

Taking (X_2, Y_2)

f $(x_2\, , y_2)$

= f (2,16) = 26

f $(x_1 + h\, , y_1 + h(x_1\, , y_1)$

= f (3, 42) = 80

$y_3 = y_2 + \frac{1}{2}$h [f $(x_2\, , y_2)$ + f $(x_2 + h\, , y_2 + h(x_2\, , y_2)$

= 16 + 0.5 [26 + 80]

$\boxed{y_3 = 69}$

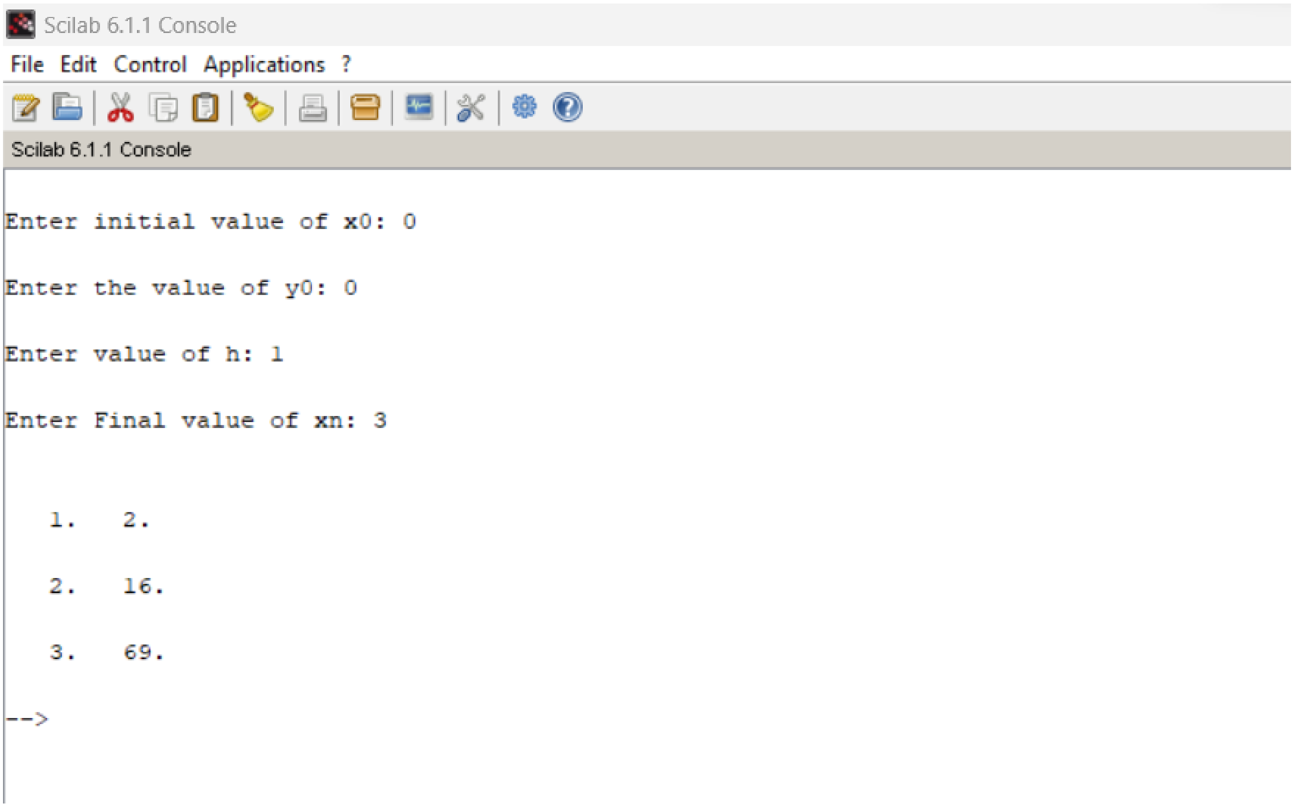
Result:

The Solution of Y(3) is 69 by using Improved Euler Method

Coding:

```
clc; clear all;
deff('g=f(x,y)','g=x^3+9*x^2-10*x+2')
x0 = input("Enter initial value of x0: ")
y0 = input("Enter the value of y0: ")
h = input("Enter value of h: ")
xn = input("Enter Final value of xn: ")
N=(xn-x0)/h
for i=1:N
   y1=y0+(h/2)*(f(x0,y0)+f(x0+h,y0+h*f(x0,y0)))
   x0=x0+h
   disp([x0 y1])
   y0=y1
   end
```

Output:


Scilab 6.1.1 Console
File Edit Control Applications ?
Scilab 6.1.1 Console
Enter initial value of x0: 0
Enter the value of y0: 0
Enter value of h: 1
Enter Final value of xn: 3
1. 2.
2. 16.
3. 69.
-->

Exercise: 20

MODIFIED EULER METHOD

Objective:

To find the solution for the given equation using Modified Euler Method.

Input:

$$y' = x^3 + 9x^2 - 10x + 2$$

Where y(x) = 0, with step length =1. To find Xn = 4

Link:

https://tutorial.math.lamar.edu/ProblemsNS/CalcII/TaylorSeries.aspx

Procedure/Methodology:

Given , $y' = x^3 + 9x^2 - 10x + 2$;

y (0) =0, h=1, y (3) =?

Here $x_0 = 0$; $y_0 = 0$; h = 1 ; $x_n = 3$

Modified Euler Formula:

$$y_{n+1} = y_n + \text{h f}\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} f(x_n, y_n)\right)$$

Taking (X_0, Y_0)

$x_0 + \frac{h}{2}$

= 0 + 0.5 = 0.5

$y_0 + \frac{h}{2} f(x_0, y_0)$

= 0 + 0.5 (2) = 1

$\text{f}(x_0 + \frac{h}{2}, y_0 + \frac{h}{2} f(x_0, y_0))$

= f (0.5,1) = - 0.625

$$y_1 = y_0 + \text{h f}\left(x_0 + \frac{h}{2}, y_0 + \frac{h}{2} f(x_0, y_0)\right)$$

= 0 + 1 (- 0.625)

$\boxed{y_1 = -0.625}$

Taking (X_1, Y_1)

$$x_1 + \frac{h}{2}$$

= 1 + 0.5 = 1.5

$$y_1 + \frac{h}{2} f(x_1, y_1)$$

= -0.625 + 0.5 (1) = 0.375

$$f\left(x_1 + \frac{h}{2}, y_1 + \frac{h}{2} f(x_1, y_1)\right)$$

= f (1.5, 0.375) = 10.625

$$y_2 = y_1 + h\, f\left(x_1 + \frac{h}{2}, y_1 + \frac{h}{2} f(x_1, y_1)\right)$$

= - 0.625 + 1 (10.625)

$$\boxed{y_2 = 10}$$

Taking (X_2, Y_2)

$$x_2 + \frac{h}{2}$$

= 2 + 0.5 = 2.5

$$y_2 + \frac{h}{2} f(x_2, y_2)$$

= 10 + 0.5 (26) = 23

$$f\left(x_2 + \frac{h}{2}, y_2 + \frac{h}{2} f(x_2, y_2)\right)$$

= f (2.5, 23) = 48.875

$$y_3 = y_2 + h\, f\left(x_2 + \frac{h}{2}, y_2 + \frac{h}{2} f(x_2, y_2)\right)$$

= 10 + 1 (48.875)

$$\boxed{y_3 = 58.875}$$

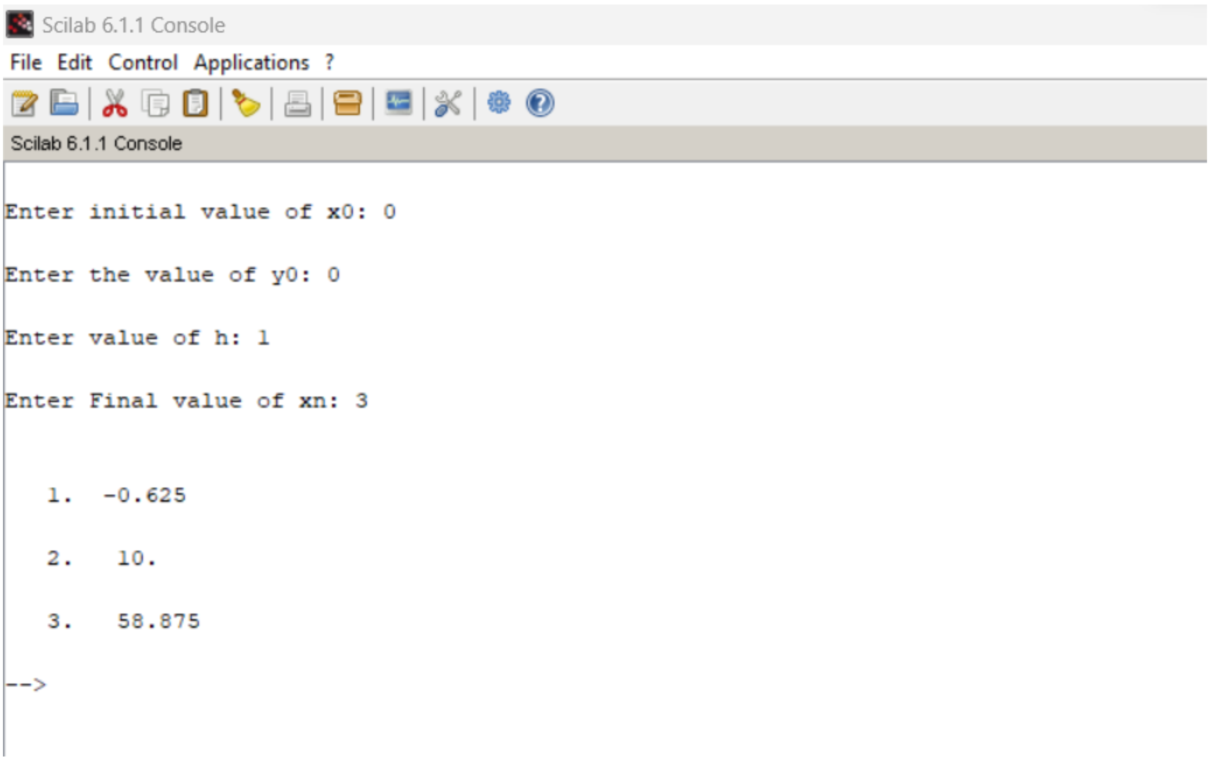
Result:

The Solution of Y(3) is 58.875 by using Modified Euler Method

Coding:

```
clc; clear all;
deff('g=f(x,y)','g=x^3+9*x^2-10*x+2')
x0 = input("Enter initial value of x0: ")
y0 = input("Enter the value of y0: ")
h = input("Enter value of h: ")
xn = input("Enter Final value of xn: ")
N=(xn-x0)/h
for i=1:N
   y1=y0+h*f(x0+(h/2),y0+(h/2)*f(x0,y0))
   x0=x0+h
   disp([x0 y1])
   y0=y1
   end
```

Output:



Exercise: 21

RUNGE KUTTA METHOD – 2^{nd} ORDER

Objective:

To find the solution for the given equation using Runge Kutta Method of Second Order.

Input:

$$y' = x^3 + 9x^2 - 10x + 2$$

Where y(x) = 0, with step length =1. To find Xn = 4

Link:

https://tutorial.math.lamar.edu/ProblemsNS/CalcII/TaylorSeries.aspx

Procedure/Methodology:

Given , $y' = x^3 + 9x^2 - 10x + 2$;

y (0) =0, h=1, y (3) =?

Here $x_0 = 0$; $y_0 = 0$; h = 1 ; $x_n = 3$

RK Formula (2nd Order):

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf(x_n + h, y_n + k_1)$$

$$y_{n+1} = y_n + \frac{k_1 + k_2}{2}$$

Taking (X_0, Y_0)

$$k_1 = hf(x_0, y_0)$$

= 1 f (0,0) = 1(2) = 2

$$k_2 = hf(x_0 + h, y_0 + k_1)$$

= 1 f (1, 2) = 1(2) = 2

$$y_1 = y_0 + \frac{k_1 + k_2}{2}$$

= 0 + 2

$\boxed{y_1 = 2}$

Taking (X_1, Y_1)

$$k_1 = hf(x_1, y_1)$$

$= 1$ f $(1,2) = 1(2) = 2$

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

$= 1$ f $(2,4) = 1(26) = 26$

$$y_2 = y_1 + \frac{k_1 + k_2}{2}$$

$= 2 + 14$

$$\boxed{y_2 = 16}$$

Taking (X_2, Y_2)

$$k_1 = hf(x_1, y_1)$$

$= 1$ f $(2,16) = 1(26) = 26$

$$k_2 = hf(x_2 + h, y_2 + k_1)$$

$= 1$ f $(3, 42) = 1(80) = 80$

$$y_3 = y_2 + \frac{k_1 + k_2}{2}$$

$= 16 + 53$

$$\boxed{y_2 = 69}$$

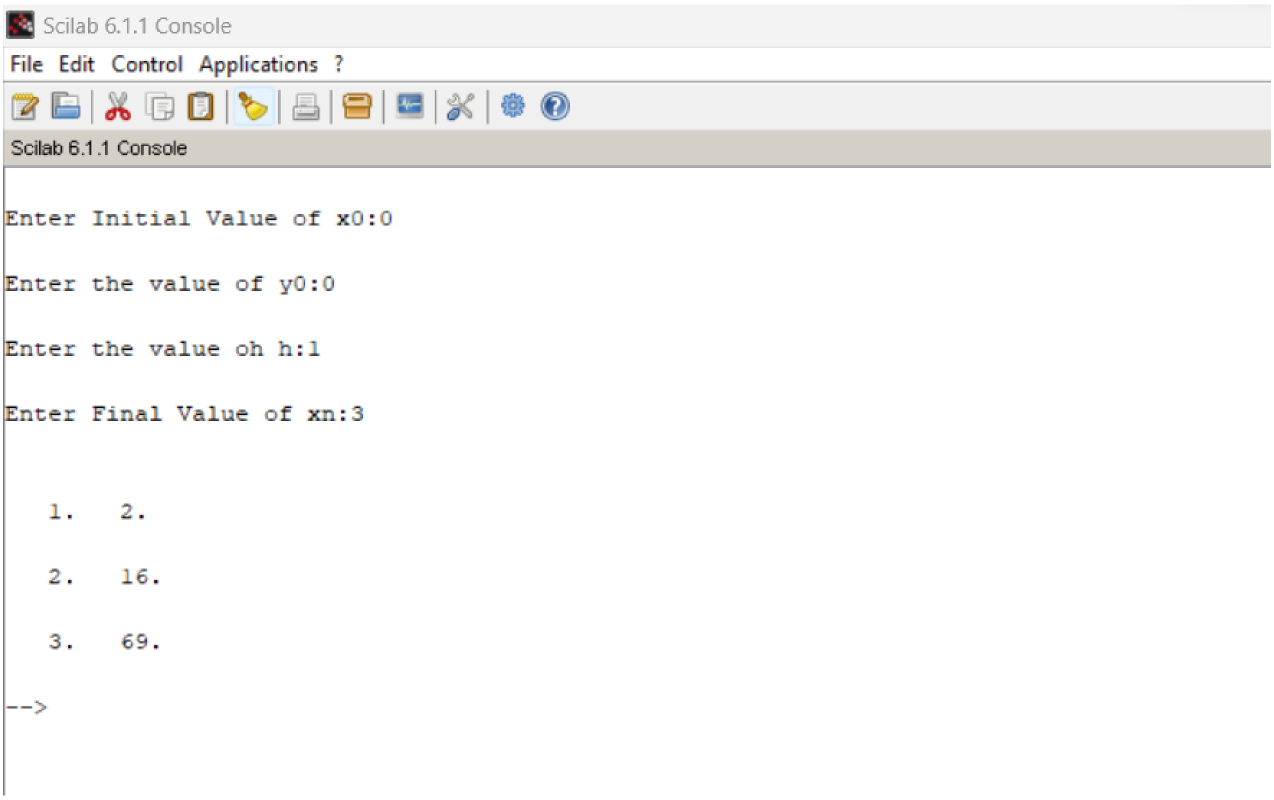
Result:

The Solution of Y(3) is 69 by using Runge Kutta Method of Second Order

Coding:

```
clc;
clear all;
deff('z=f(x,y)','z=x^3+9*x^2-10*x+2')
x0=input("Enter Initial Value of x0:")
y0=input("Enter the value of y0:")
h=input("Enter the value oh h:")
xn=input("Enter Final Value of xn:")
n=(xn-x0)/h
for i=1:n
   k1=h*f(x0,y0)
   k2=h*f(x0+h,y0+k1)
   y1=y0+0.5*(k1+k2)
   x0=x0+h
   disp([x0 y1])
   y0=y1
end
```

Output:



Exercise: 21

RUNGE KUTTA METHOD – 3^{rd} ORDER

Objective:

To find the solution for the given equation using Runge Kutta Method of Third Order.

Input:

$$y' = x^3 + 9x^2 - 10x + 2$$

Where y(x) = 0, with step length =1. To find Xn = 4

Link:

https://tutorial.math.lamar.edu/ProblemsNS/CalcII/TaylorSeries.aspx

Procedure/Methodology:

Given , $y' = x^3 + 9x^2 - 10x + 2$;

y (0) =0, h=1, y (3) =?

Here x_0= 0 ; y_0= 0 ; h = 1 ; x_n= 3

RK Formula (3rd Order):

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$

$$k_3 = hf(x_n + h, y_n + 2k_2 - k_1)$$

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 4k_2 + k_3)$$

Taking (X_0, Y_0)

$$k_1 = hf(x_0, y_0)$$

= 1 f (0,0) = 1(2) = 2

$$k_2 = hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}\right)$$

= 1 f (0.5, 1) = 1(- 0.625) = -0.625

$$k_3 = hf(x_n + h, y_n + 2k_2 - k_1)$$

= 1 f (1, -3.25) = 1(2) = 2

$$y_1 = y_0 + \frac{1}{6}\,(k_1 + 4k_2 + \ k_3)$$

$= 0 + 1/6 \,(2 + 4(-0.625) +2)$

$\boxed{y_1 = 0.25}$

Taking (X_1, Y_1)

$k_1 = hf(x_1, y_1)$

$= 1$ f $(1, 0.25) = 1(2) = 2$

$k_2 = hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}\right)$

= 1 f (1.5, 1.25) = 1(10.625) = 10.625

$k_3 = hf(x_1 + h, y_1 + \ 2k_2 - k_1\ )$

$= 1$ f $(2, 19.5) = 1(26) = 26$

$$y_2 = y_1 + \frac{1}{6}\,(k_1 + 4k_2 + \ k_3)$$

$= 0.25 + 1/6 \,(2 + 4(10.625) +26)$

$\boxed{y_2 = 12}$

Taking (X_2, Y_2)

$k_1 = hf(x_2, y_2)$

$= 1$ f $(2, 12) = 1(26) = 26$

$k_2 = hf\left(x_2 + \frac{h}{2}, y_2 + \frac{k_1}{2}\right)$

= 1 f (2.5, 25) = 1(48.875) = 48.875

$k_3 = hf(x_1 + h, y_1 + \ 2k_2 - k_1\ )$

$= 1$ f $(3, 83.75) = 1(80) = 80$

$$y_3 = y_2 + \frac{1}{6}\,(k_1 + 4k_2 + \ k_3)$$

$= 12 + 1/6 \,(26 + 4(48.875) +80)$

$\boxed{y_3 = 62.25}$

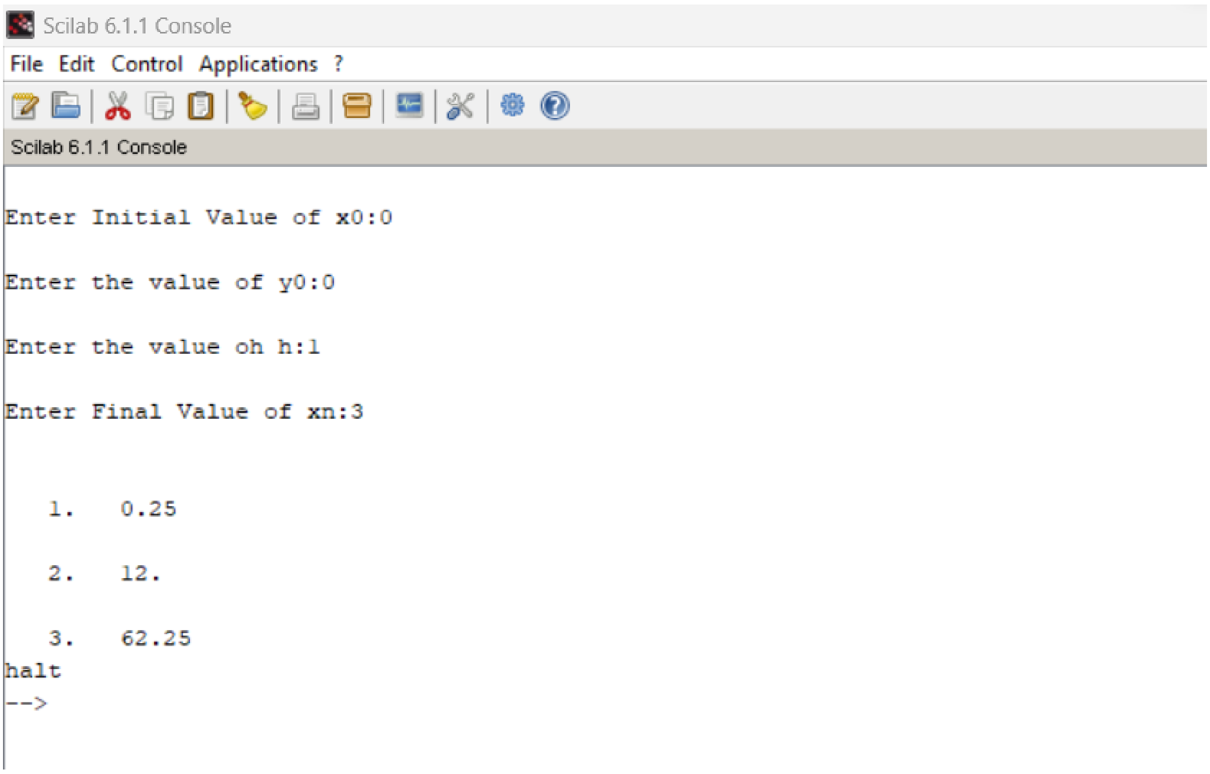
Result:

The Solution of Y(3) is 62.25 by using Runge Kutta Method of Third Order

Coding:

```
clc;
clear all;
deff('z=f(x,y)','z=x^3+9*x^2-10*x+2')
x0=input("Enter Initial Value of x0:")
y0=input("Enter the value of y0:")
h=input("Enter the value oh h:")
xn=input("Enter Final Value of xn:")
n=(xn-x0)/h
for i=1:n
   k1=h*f(x0,y0)
   k2=h*f(x0+(h/2),y0+(k1/2))
   k3=h*f(x0+h,y0+2*k2-k1)
   y1=y0+(1/6)*(k1+4*k2+k3)
   x0=x0+h
   disp([x0 y1])
   y0=y1
end
halt( )
```

Output:



Exercise: 21 **Date:**

RUNGE KUTTA METHOD – 4 rd ORDER

Objective:

To find the solution for the given equation using Runge Kutta Method of Fourth Order.

Input:

$$y' = x^3 + 9x^2 - 10x + 2$$

Where y(x) = 0, with step length =1. To find Xn = 4

Link:

https://tutorial.math.lamar.edu/ProblemsNS/CalcII/TaylorSeries.aspx

Procedure/Methodology:

Given , $y' = x^3 + 9x^2 - 10x + 2$;

y (0) =0, h=1, y (3) =?

Here x_0= 0 ; y_0= 0 ; h = 1 ; x_n= 3

RK Formula (3rd Order):

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$

$$k_3 = hf\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right)$$

$$k_4 = hf(x_n + h, y_n + k_3)$$

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Taking (X_0, Y_0)

$$k_1 = hf(x_0, y_0)$$

= 1 f (0,0) = 1(2) = 2

$$k_2 = hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}\right)$$

= 1 f (0.5, 1) = 1(- 0.625) = -0.625

$$k_3 = hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}\right)$$

= 1 f (0.5, -0.3125) = 1(- 0.625) = -0.625

$k_4 = hf(x_0 + h, y_0 + k_3\ )$

= 1 f (1, -0.625) = 1(2) = 2

$$y_1 = y_0 + \frac{1}{6}\ (\ k_1 + 2k_2 + 2k_3 +\ k_4)$$

= 0 + 1/6 (2 + 2(-0.625) + 2(-0.625) + 2)

$\boxed{y_1 = 0.25}$

Taking (X_1, Y_1)

$k_1 = hf(x_1, y_1)$

= 1 f (1, 0.25) = 1(2) = 2

$k_2 = hf\left(x_1 + \frac{h}{2}, y_1 + \frac{k_1}{2}\right)$

= 1 f (1.5, 1.25) = 1(10.625) = 10.625

$k_3 = hf\left(x_1 + \frac{h}{2}, y_1 + \frac{k_2}{2}\right)$

= 1 f (1.5, 5.5625) = 1(10.625) = 10.625

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

= 1 f (2, 10.625) = 1(26) = 26

$$y_2 = y_1 + \frac{1}{6}\ (\ k_1 + 2k_2 + 2k_3 +\ k_4)$$

= 0.25 + 1/6 (2 + 2(10.625) + 2(10.625) + 26)

$\boxed{y_2 = 26}$

Taking (X_2, Y_2)

$k_1 = hf(x_2, y_2)$

= 1 f (2, 12) = 1(26) = 26

$k_2 = hf\left(x_2 + \frac{h}{2}, y_2 + \frac{k_1}{2}\right)$

= 1 f (2.5, 25) = 1(48.875) = 48.875

$k_3 = hf\left(x_1 + \frac{h}{2}, y_1 + \frac{k_2}{2}\right)$

= 1 f (2.5, 36.4375) = 1(48.875) = 48.875

$k_4 = hf(x_0 + h, y_0 + k_3\ )$

$$= 1\ f\ (\ 3, 60.875) = 1(80) = 80$$

$$y_3 = y_2 + \frac{1}{6}\ (\ k_1 + 2k_2 + 2k_3 + \ k_4)$$

$$= 26 + 1/6\ (\ 26 + 2(48.875) + 2(48.875) + 80)$$

$$\boxed{y_3 = 62.25}$$

Result:

The Solution of Y(3) is 62.25 by using Runge Kutta Method of Fourth Order

Coding:

```
clc;
clear all;
deff('z=f(x,y)','z=x^3+9*x^2-10*x+2')
x0=input("Enter Initial Value of x0:")
y0=input("Enter the value of y0:")
h=input("Enter the value of h:")
xn=input("Enter Final Value of xn:")
n=(xn-x0)/h
for i=1:n
   k1=h*f(x0,y0)
   k2=h*f(x0+(h/2),y0+(k1/2))
   k3=h*f(x0+h,y0+(k2/2))
   k4=h*f(x0+h,y0+k3)
   y1=y0+(1/6)*(k1+2*k2+2*k3+k4)
   x0=x0+h
   disp([x0 y1])
   yo=y1
end
```

Output:

Scilab 6.1.1 Console

File Edit Control Applications ?

Scilab 6.1.1 Console

```
Enter Initial Value of x0:0

Enter the value of y0:0

Enter the value of h:1

Enter Final Value of xn:3


   1.   1.125

   2.   16.875

   3.   60.625
```

Exercise: 21

ADAMS BASH FORTH PREDICTOR CORRECTOR METHOD

Objective:

To find the solution for the given equation using Adams Bash forth Predictor Corrector Method

Input:

$y' = x^2 + \frac{Y}{2}$; To find X = 1.4

X	1.0	1.1	1.2	1.3
Y	2.0	2.2156	2.4649	2.7514

Link:

Procedure/Methodology:

Adams Bash Forth Predictor Formula:

$$y_{n+1,P} = y_n + \frac{h}{24}[55y'_n - 59y'_{n-1} + 37y'_{n-2} - 9y'_{n-3}]$$

Adams Bash Forth Corrector Formula:

$$y_{n+1,C} = y_n + \frac{h}{24}[9y'_{n+1} + 19y'_n - 5y'_{n-1} - y'_{n-2}]$$

Given, $y' = x^2 + \frac{Y}{2}$

When x = 1 ; y = 2

$$Y_0{}' = 1^2 + \frac{2}{2} \qquad =2$$

When x = 1.1 ; y = 2.2156

$$Y_1{}' = (1.1)^2 + \frac{2.2156}{2} \qquad = 2.3178$$

When x = 1.2 ; y = 2.4649

$$Y_2{}' = (1.2)^2 + \frac{2.4649}{2} \qquad = 2.6724$$

When x = 1 ; y = 2

$$Y_3{}' = (1.3)^2 + \frac{2.7514}{2} \qquad = 3.0657$$

Putting n = 3 in Adams Bash Forth Predictor Formula:

$$y_{1.4,P} = y_3 + \frac{h}{24}[55y_3' - 59y_2' + 37y_1' - 9y_0']$$

$y_{1.4,P}$ = 2.7514 + 0.1/24 [55(3.0657) -59(2.6724) +37(2.3178) -9(2)]

$y_{1.4,P}$ = 2.7514 + 0.1/24 [78.6976]

$\boxed{y_{1.4,P} = 3.0793}$

When x = 1.4 ; y = 3.0793

$$Y_4{}' = (1.4)^2 + \frac{3.0793}{2} \qquad = 3.4997$$

Putting n = 3 in Adams Bash Forth Corrector Formula:

$$y_{1.4,C} = y_3 + \frac{h}{24}[9y_4' + 19y_3' - 5y_2' - y_1']$$

$y_{1.4,C}$ = 2.7514 + 0.1/24 [9(3.4997) 19(3.0657) -5(2.6724 +12(2.3178)]

$y_{1.4,C}$ = 2.7514 + 0.1/24 [78.7007]

$\boxed{y_{1.4,C} = 3.0793}$

Result:

The Solution of Y(1.4) is 3.0793 by using Adams Bash Forth Predictor and Corrector Method.

Coding:

```
//Adam's Method
clc; clear all;
disp("Adams Bashforth Predictor Corrector Method")
deff('g=f(x,y)','g=x^2+(y/2)')
x0=1; y0=2; xn=1.3; xf=1.4;
h=0.1;
N=(xn-x0)/h
function [x, y]=Adam(f, x0, y0, h, N)
   x = zeros(N+1, 1)
   y = zeros(N+1,1)
   x(1) = x0
   y(1) = y0
   for i=1:N
      k1=h*f(x(i),y(i))
      k2=h*f(x(i)+(h/2),y(i)+(k1/2))
```

```
        k3=h*f(x(i)+(h/2),y(i)+(k2/2))
        k4=h*f(x(i)+h,y(i)+k3)
        y(i+1)=y(i)+(1/6)*(k1+2*k2+2*k3+k4)
        x(i+1) = x(i) + h
    end
    yo=f(x(1),y(1))
    y1=f(x(2),y(2))
    y2=f(x(3),y(3))
    y3=f(x(4),y(4))
    p=y(4)+(h/24)*(55*y3-59*y2+37*y1-9*yo)
    printf("Predicted Value is %f",p)
    y4=f(xf,p)
    printf("\nAt x= 0.4 and y = %f then y4 = %f",p,y4)
    c=y(4)+(h/24)*(9*y4+19*y3-5*y2+y1)
    printf("\nCorrector Value is %f",c)
    printf("\n\nBy Help of Using Runge-Kutta 4th Order\n")
    printf("\n\tx and y values\n")
endfunction
```

Output:

```
Scilab 6.1.0 Console
File  Edit  Control  Applications  ?
Scilab 6.1.0 Console

  "Adams Bashforth Predictor Corrector Method"

--> [x, y]=Adam(f, x0, y0, h, N)
Predicted Value is 3.079274
At x= 0.4 and y = 3.079274 then y4 = 3.499637
Corrector Value is 3.079277

By Help of Using Runge-Kutta 4th Order

        x and y values
 x  =

   1.
   1.1
   1.2
   1.3
 y  =

   2.
   2.2155907
   2.4647857
   2.7513587

-->
```

Exercise: 21 **Date:**

MILNE PREDICTOR CORRECTOR METHOD

Objective:

To find the solution for the given equation using Milne Predictor Corrector Method

Input:

$y' = x^2 + \frac{Y}{2}$; To find X = 1.4

X	1.0	1.1	1.2	1.3
Y	2.0	2.2156	2.4649	2.7514

Link:

Procedure/Methodology:

Milne Predictor Formula:

$$y_{n+1,P} = y_{n-3} + \frac{4h}{24}[2y'_n - y'_{n-1} + 2y'_{n-2}]$$

Milne Forth Corrector Formula:

$$y_{n+1,C} = y_{n-1} + \frac{h}{3}[y'_{n-1} + 4y'_n + y'_{n+1}]$$

Given, $y' = x^2 + \frac{Y}{2}$

When x = 1 ; y = 2

$$Y_0{}' = 1^2 + \frac{2}{2} \qquad =2$$

When x = 1.1 ; y = 2.2156

$$Y_1{}' = (1.1)^2 + \frac{2.2156}{2} \qquad = 2.3178$$

When x = 1.2 ; y = 2.4649

$$Y_2{}' = (1.2)^2 + \frac{2.4649}{2} \qquad = 2.6724$$

When x = 1 ; y = 2

$$Y_3{}' = (1.3)^2 + \frac{2.7514}{2} \qquad = 3.0657$$

Putting n = 3 in Adams Bash Forth Predictor Formula:

$$y_{1.4,P} = y_0 + \frac{4h}{24}[2y_1' - y_2' + 2y_3']$$

$y_{1.4,P}$ = 2+ 4(0.1)/24 [2(2.3178) – 2.6724 + 2(3.0657)]

$y_{1.4,P}$ = 2.7514 + 0.4/24 [8.0946]

$\boxed{y_{1.4,P} = 3.0793}$

When x = 1.4 ; y = 3.0793

$$Y_4{}' = (1.4)^2 + \frac{3.0793}{2} \qquad = 3.4997$$

Putting n = 3 in Adams Bash Forth Corrector Formula:

$$y_{1.4,C} = y_3 + \frac{h}{3}[y_2' + 4y_3' + y_4']$$

$y_{1.4,C}$ = 24649 + 0.1/3 [2.6724 + 4(3.0657) + 3.4996]

$y_{1.4,C}$ = 2.4649 + 0.1/3 [18.4349]

$\boxed{y_{1.4,C} = 3.0793}$

Result:

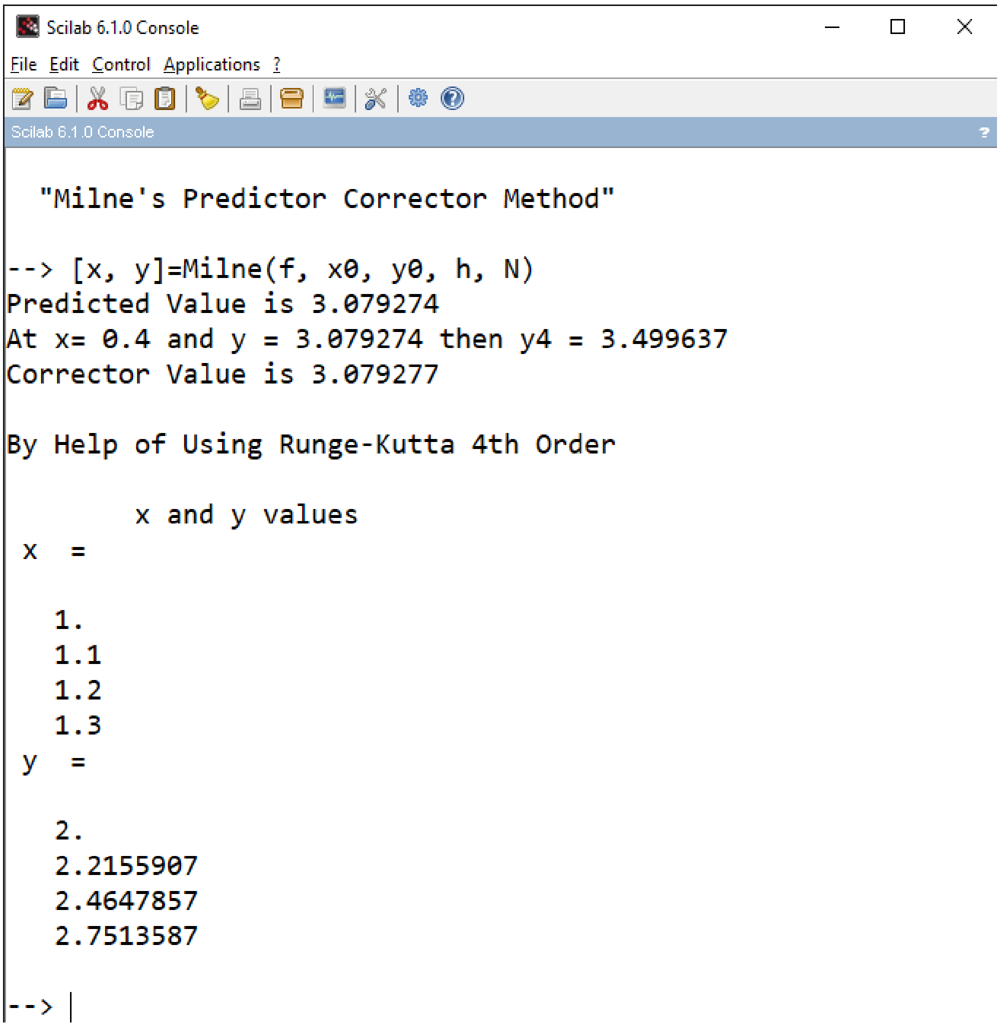
The Solution of Y(1.4) is 3.0793 by using Adams Bash Forth Predictor and Corrector Method.

Coding:

```
//Milne's Method
clc; clear all;
disp("Milne"s Predictor Corrector Method")
deff('g=f(x,y)','g=x^2+(y/2)')
x0=1; y0=2; xn=1.3; xf=1.4;
h=0.1;
N=(xn-x0)/h
function [x, y]=Milne(f, x0, y0, h, N)
   x = zeros(N+1, 1)
   y = zeros(N+1,1)
   x(1) = x0
   y(1) = y0
   for i=1:N
      k1=h*f(x(i),y(i))
      k2=h*f(x(i)+(h/2),y(i)+(k1/2))
      k3=h*f(x(i)+(h/2),y(i)+(k2/2))
```

```
        k4=h*f(x(i)+h,y(i)+k3)
        y(i+1)=y(i)+(1/6)*(k1+2*k2+2*k3+k4)
        x(i+1) = x(i) + h
    end
    y1=f(x(2),y(2))
    y2=f(x(3),y(3))
    y3=f(x(4),y(4))
    p=y(1)+(4*h/3)*(2*y1-y2+2*y3)
    printf("Predicted Value is %f",p)
    y4=f(xf,p)
    printf("\nAt x= 0.4 and y = %f then y4 = %f",p,y4)
    c=y(3)+(h/3)*(y2+4*y3+y4)
    printf("\nCorrector Value is %f",c)
    printf("\n\nBy Help of Using Runge-Kutta 4th Order\n")
    printf("\n\tx and y values\n")
endfunction
```

Output:



MODULE

5

Exercise: 26

LEIBMANN'S METHOD

Objective:

To find the solution for the given Partial Differential Equation using Liebman's Method.

Input:

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

Boundary Value:

0, 1000, 2000, 1000, 0, 500, 1000, 500, 0, 1000, 2000, 1000, 0, 500, 1000, 500.

Procedure/Methodology:

Solve the Elliptic equation $u_{xx} + u_{yy} = 0$ for the following square mesh with boundary value as shown in figure.

Solution:

Let $u_1, u_2, u_3, \ldots, u_9$ be the value of u at the interior mesh – points sine the boundary value of u are symmetrical about x-axis

$u_1 = u_7, u_3 = u_9, u_8 = u_2$

Also, Symmetrical about y – axis

$u_1 = u_3, u_4 = u_6, u_9 = u_7$

Now, find the value u_1, u_2, u_4, u_5

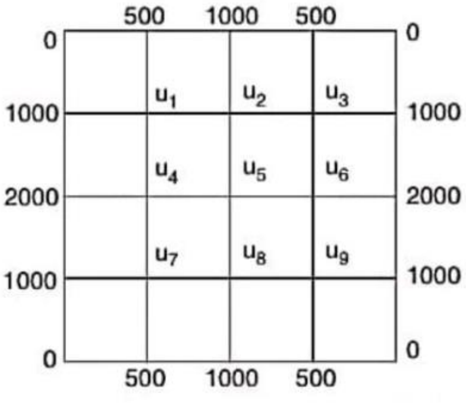


Figure 26.1: Boundary value with 16 square mesh

Initial Iteration:

$u_5^{(0)} = \frac{1}{4}$ [1000+1000+2000+2000] = 1500(SFPF)

$u_1^{(0)} = \frac{1}{4}$ [0+1500+2000+1000] = 1125(DFPF)

$u_2^{(0)} = \frac{1}{4}$ [1125+1125+1500+1000] = 1188(SFPF)

$u_4^{(0)} = \frac{1}{4}$ [2000+1500+1125+1125] = 1438(SFPF)

By Standard Five Point Formula

$u_1^{(n+1)} = \frac{1}{4}$ [1000+ $u_2^{(n)}$ + 500 + $u_4^{(n)}$]

$u_2^{(n+1)} = \frac{1}{4}$ [$u_1^{(n+1)}$ + $u_1^{(n)}$ + 1000 + $u_5^{(n)}$]

$u_4^{(n+1)} = \frac{1}{4}$ [2000+ $u_5^{(n)}$ + $u_1^{(n+1)}$ + $u_1^{(n)}$]

$u_5^{(n+1)} = \frac{1}{4}$ [$u_1^{(n+1)}$ + $u_4^{(n)}$ + $u_2^{(n+1)}$ + $u_2^{(n)}$]

Iteration 1:

$u_1^{(1)} = \frac{1}{4}$ [1000+1188+500+1438] = 1032

$u_2^{(1)} = \frac{1}{4}$ [1032+1125+1000+1500] = 1164

$u_4^{(1)} = \frac{1}{4}$ [2000+1500+1032+1125] = 1414

$u_5^{(1)} = \frac{1}{4}$ [1414+1438+1164+1188] = 1301

Iteration 2:

$u_1^{(2)} = \frac{1}{4}$ [1000+1164+500+1414] = 1020

$u_2^{(2)} = \frac{1}{4}$ [1020+1032+1000+1301] = 1088

$u_4^{(2)} = \frac{1}{4}$ [2000+1301+10201032] = 1338

$u_5^{(2)} = \frac{1}{4}$ [1338+1414+1088+1164] = 1251

Iteration	u_1	u_2	u_4	u_5
3	982	1063	1313	1201
4	969	1038	1288	1176
5	957	1026	1276	1157
6	951	1016	1266	1146
7	946	1011	1260	1138
8	943	1007	1257	1134
9	941	1005	1255	1131
10	940	1003	1253	1129
11	939	1002	1252	1128
12	939	101	1251	1126

Coding:

```
clc;
clear all;
printf("Leibmann"s Method")
b=[0,1000,2000,1000,0,500,1000,500,0,1000,2000,1000,0,500,1000,500];
a=1/4;
printf("\nSymmetrical Problem")
printf("\nInitial Value (Iteration)")
u5=a*(b(3)+b(11)+b(7)+b(15))
u1=a*(b(1)+u5+b(3)+b(15))
u2=a*(2*u1+u5+b(15))
u4=a*(b(3)+u5+2*u1)
printf("\nDiagonal - Five Point Formul")
disp(u1,'u1(0)')
printf("\nStandard - Five Point Formul")
disp(u2,'u2(0)')
disp(u4,'u4(0)')
disp(u5,'u5(0)')
for i=1:11
   u11=a*(b(2)+u2+b(16)+u4)
   u21=a*(u11+u1+u5+b(15))
   u41=a*(b(3)+u5+u1+u11)
   u51=a*(u21+u2+u41+u4)
   u1=u11;u2=u21;u4=u41;u5=u51;
   printf('\tu1(%i)=%g\n\tu2(%i)=%g\n\tu4(%i)=%g\n\tu5(%i)=%g\n\t\n',i,u1,i,u2,i,u4,i,u5)
end
printf('Solution of converges Laplace"s Equation to \n\t%g,%g,%g and %g',u1,u2,u4,u4)
```

Output:

```
Scilab 6.1.1 Console
File  Edit  Control  Applications  ?
Scilab 6.1.1 Console
Leibmann's Method
Symmetrical Problem
Initial Value (Iteration)
Diagonal - Five Point Formul
   1125.

  "u1(0)"

Standard - Five Point Formul
   1187.5

  "u2(0)"

   1437.5

  "u4(0)"

   1500.

  "u5(0)"
        u1(1)=1031.25
        u2(1)=1164.06
        u4(1)=1414.06
        u5(1)=1300.78

        u1(2)=1019.53
        u2(2)=1087.89
        u4(2)=1337.89
        u5(2)=1250.98

        u1(3)=981.445
        u2(3)=1062.99
        u4(3)=1312.99
        u5(3)=1200.44

        u1(4)=968.994
        u2(4)=1037.72
        u4(4)=1287.72
        u5(4)=1175.35
```

```
Scilab 6.1.1 Console
File  Edit  Control  Applications  ?
Scilab 6.1.1 Console

        u1(5)=956.36
        u2(5)=1025.18
        u4(5)=1275.18
        u5(5)=1156.45

        u1(6)=950.089
        u2(6)=1015.72
        u4(6)=1265.72
        u5(6)=1145.45

        u1(7)=945.362
        u2(7)=1010.23
        u4(7)=1260.23
        u5(7)=1137.97

        u1(8)=942.613
        u2(8)=1006.49
        u4(8)=1256.49
        u5(8)=1133.36

        u1(9)=940.744
        u2(9)=1004.18
        u4(9)=1254.18
        u5(9)=1130.33

        u1(10)=939.589
        u2(10)=1002.67
        u4(10)=1252.67
        u5(10)=1128.42

        u1(11)=938.833
        u2(11)=1001.71
        u4(11)=1251.71
        u5(11)=1127.19

Solution of converges Laplace's Equation to
        938.833,1001.71,1251.71 and 1251.71
-->
```

Exercise: 27

POISSON'S METHOD

Objective:

To find the solution for the given Partial Differential Equation using Poisson's Method.

Input:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 8x^2y^2$$

Boundary Value:

u (x, y) = 0

Procedure/Methodology:

Solve the Elliptic equation $u_{xx} + u_{yy} = 8x^2\ y^2$ for the following square mesh with u(x, y) = 0 on the boundaries dividing the square into 16 sub – squares of length 1 unit (show in figure below)

Solution:

$$u_{i-1\,,j} + u_{i+1\,,j} + u_{i\,,j-1} + u_{i,j-1} - 4\,u_{i,j} = 8i^2\ j^2$$

Here Boundary values are symmetrical about x-axis and y-axis

$u_1 = u_3 = u_7 = u_9$ and

$u_2 = u_4 = u_6 = u_8$

Now, find the value u_1, u_2, and u_5

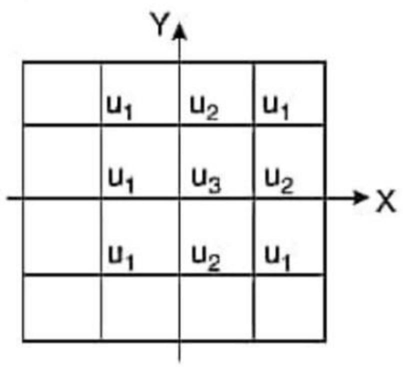


Figure 27.1: Boundary value with 16 square mesh

At u_3(i =1, j=1)

$$u_{o,1} + u_{2,1} + u_{1,0} + u_{1,0} - 4u_{1,1} = 8$$

$$u_2 + 0 + 0 + u_2 - 4u_3 = 8$$

$$2\,u_2 - 4\,u_1 = 8 \qquad \text{------(1)}$$

At u_5(i =0, j=0)

$$u_{-1,0} + u_{1,0} + u_{0,1} + u_{0,-1} - 4u_{0,0} = 0$$

$$u_4 + u_6 + u_8 + u_2 - 4u_5 = 0$$

$$u_2 = u_5 \qquad \text{------(2)}$$

At u_2(i =0, j=1)

$$u_{-1,1} + u_{1,1} + u_{0,2} + u_{0,0} - 4u_{0,1} = 0$$

$$u_1 + u_1 + 0 + u_5 - 4u_2 = 0$$

$$2u_2 + u_5 = 4u_2 \qquad \text{------(3)}$$

From equation (2) and (3)

$$(2) \times 2 \Rightarrow \quad \cancel{4u_1} - 6u_2 = 0$$

$$(1) \times 1 \Rightarrow \quad -\cancel{4u_1} + 2u_2 = 0$$

$$-4\,u_2 = 8$$

$$u_2 = -2$$

Substitution in equation (1), We get

$$u_1 = -3$$

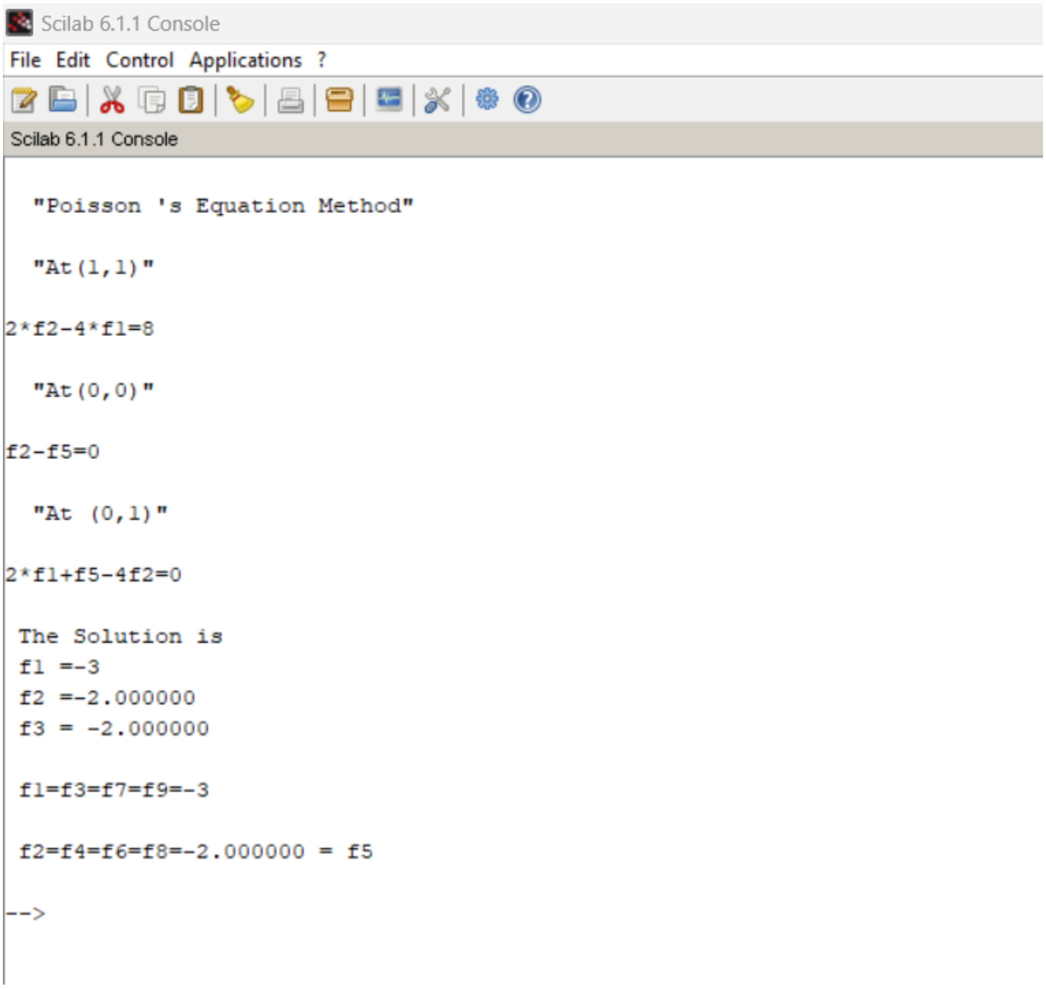
$u_1 = u_3 = u_7 = u_9 = -3$

$u_2 = u_4 = u_6 = u_8 = u_5 = -2$

Coding:

```
clc;
clear all;
disp("Poisson "s Equation Method")
deff("z=f(x,y)","z=8*x^2*y^2");
b=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
disp("At(1,1)")
printf("\t\n2*f2-4*f1=8\n")
disp("At(0,0)")
printf("\t\nf2-f5=0\n")
disp("At (0,1)")
printf("\t\n2*f1+f5-4f2=0\n")
A=[-4,2,0;0,1,-1;2,-4,1]
B=[8,0,0]
C= inv(A)*B';
mprintf('\n The Solution is \n f1 =%d \n f2 =%0f \n f3 = %0f \n',C(1),C(2),C(3))
printf('\t\n f1=f3=f7=f9=%d\n',C(1))
printf('\t\n f2=f4=f6=f8=%0f = f5\n',C(3))
```

Output:



Exercise: 28

BENDER - SCHMIDT METHOD

Objective:

To find the solution for the given Partial Differential Equation using Bender - Schmidt Method.

Input:

$$\frac{\partial^2 u}{\partial x^2} = 2\frac{\partial u}{\partial t}$$

Boundary Conditions :

u (0, t) = u (4, t) = 0

u (x, 0) = x (4 – x)

Procedure/Methodology:

Solve u_{xx} = 2 u_t under conditions u (0, t) = u(4, t) = 0 and u (x, 0) = x(4 – x) taking h =1 find the value up to t = 5

Solution:

$$u_{xx} = 2u_t$$
$$\alpha = 2, h = 1$$
$$k = \frac{\alpha h^2}{2} = 1$$
$$\lambda = \frac{k}{\alpha h^2} = \frac{1}{2(1)^2}$$
$$\lambda = \frac{1}{2}$$

General Formula

$u_{(i,j+1)}$ = ½ [$u_{(i+1), j}$ + $u_{(i-1,j)}$]

u (x, 0) = x (4 – x)

u (1, 0) = 1 (4 -1) = 3

u (2, 0) = 2 (4 – 2) = 4

u (3, 0) = 3(4 – 3) = 3

For j = 0

$u_{i,1}$ = ½ [$u_{i+1,0}$ + $u_{i-1,0}$]

$u_{1,1}$ = ½ [$u_{2,0}$ + $u_{0,0}$] = 2

$u_{2,1}$ = ½ [$u_{3,0}$ + $u_{1,0}$] = 3

$u_{3,1}$ = ½ [$u_{4,0}$ + $u_{2,0}$] = 2

For j = 1

$u_{i,2} = ½ [u_{i+1,1} + u_{i-1,1}]$

$u_{1,2} = ½ [u_{2,1} + u_{0,1}] = 1.5$

$u_{2,2} = ½ [u_{3,1} + u_{1,1}] = 2$

$u_{3,2} = ½ [u_{4,1} + u_{2,1}] = 1.5$

For j = 2

$u_{i,3} = ½ [u_{i+1,2} + u_{i-1,2}]$

$u_{1,3} = ½ [u_{2,2} + u_{0,2}] = 1$

$u_{2,3} = ½ [u_{3,2} + u_{1,2}] = 1.5$

$u_{3,3} = ½ [u_{4,2} + u_{2,2}] = 1$

For j = 3

$u_{i,4} = ½ [u_{i+1,4} + u_{i-1,4}]$

$u_{1,4} = ½ [u_{2,4} + u_{0,4}] = 0.75$

$u_{2,4} = ½ [u_{3,4} + u_{1,4}] = 1$

$u_{3,4} = ½ [u_{4,4} + u_{2,4}] = 0.75$

For j = 4

$u_{i,5} = ½ [u_{i+1,5} + u_{i-1,5}]$

$u_{1,5} = ½ [u_{2,5} + u_{0,5}] = 0.5$

$u_{2,5} = ½ [u_{3,5} + u_{1,5}] = 0.75$

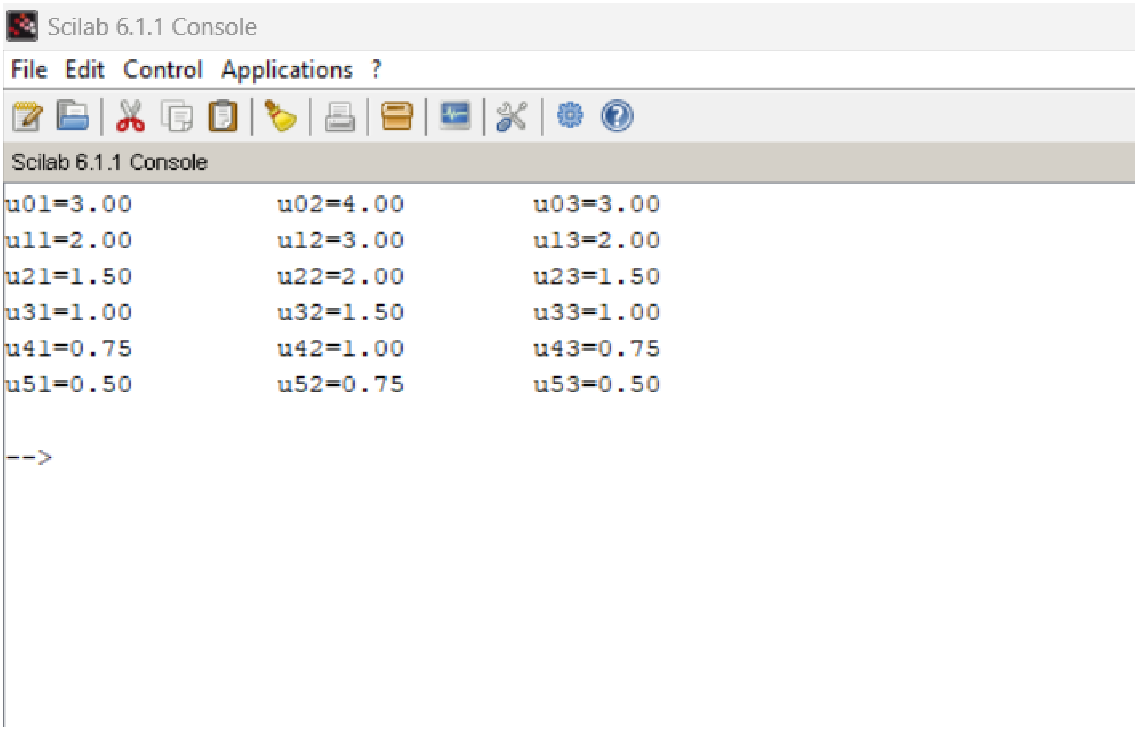
$u_{3,5} = ½ [u_{4,5} + u_{2,5}] = 0.5$

j \ i	**0**	**1**	**2**	**3**	**4**
0	0	3	4	3	0
1	0	2	3	2	0
2	0	1.5	2	1.5	0
3	0	1	1.5	1	0
4	0	0.75	1	0.75	0
5	0	0.5	0.75	0.5	0

Coding:

```
clc;
clear all;
deff('y=f(x)','y=4*x-x^2')
u=[f(0) f(1) f(2) f(3) f(4)];
b=0
printf('u01=%0.2f\t u02=%0.2f\t u03=%0.2f\t \n',f(1),f(2),f(3) )
u11=(u(1)+u(3))/2;
u12=(u(2)+u(4))/2;
u13=(u(3)+u(5))/2;
printf('u11=%0.2f\t u12=%0.2f\t u13=%0.2f\t \n',u11,u12,u13)
u21=(u(1)+u12)/2;
u22=(u11+u13)/2;
u23=(u12+0)/2;
printf('u21=%0.2f\t u22=%0.2f\t u23=%0.2f\t \n',u21,u22,u23)
u31=(u(1)+u22)/2;
u32=(u21+u23)/2;
u33=(u22-u(1))/2;
printf('u31=%0.2f\t u32=%0.2f\t u33=%0.2f\t \n',u31,u32,u33)
u41=(u(1)+u32)/2;
u42=(u33+u33)/2;
u43=(u32+u(1))/2;
printf('u41=%0.2f\t u42=%0.2f\t u43=%0.2f\t \n',u41,u42,u43)
u51=(u(1)+u42)/2;
u52=(u43+u43)/2;
u53=(u42+u(1))/2;
printf('u51=%0.2f\t u52=%0.2f\t u53=%0.2f\t \n',u51,u52,u53)
```

Output:



Exercise: 29

CRANK - NICHOLSON METHOD

Objective:

To find the solution for the given Partial Differential Equation using Crank – Nicholson Method.

Input:

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t}$$

Boundary Conditions :

u (0, t) = u (1, t) = 0

u (x, 0) = 100 (x – x^2)

Procedure/Methodology:

Solve $u_{xx} = u_t$ under conditions u (0, t) = u(1, t) = 0 and u (x, 0) = 100(x – x^2) taking h =0.25 for one time step

Solution:

a = 1 , h = 0.25

k = ah^2 = $1(0.25)^2$ = 0.0625

u (x, 0) = 100 (x – x^2)

u (0.25, 0) = 100(0.25 – 0.25^2) = 18.75

u (0.5, 0) = 100(0.5 – 0.5^2) = 25

u (0.75, 0) = 100(0.75 – 0.75^2) = 18.75

General Formula

$u_{(i,j+1)} = ¼\ [\ u_{(i+1,j+1)} + u_{(i-1,j+1)} + u_{(i-1,j)} + u_{(i+1,j)}$

For (i=1, j=0)

$u_{1,0} = ¼\ [u_{2,1} + u_{0,1} + u_{0,0} + u_{2,0}]$

$u_1 = ¼\ [u_2 + 0 + 0 + 25]$

$4u_1 – u_2 = 25$ ------(1)

For (i=2, j=0)

$u_{2,1} = ¼\ [u_{3,1} + u_{1,1} + u_{1,0} + u_{3,0}]$

$u_2 = ¼\ [18.75 + 18.75 + u_2 + u_2]$

$4u_1 – u_2 – u_3 = 37.5$ ------(2)

For (i=3, j=0)

$u_{3,1} = ¼ [u_{4,1} + u_{2,1} + u_{2,0} + u_{4,0}]$

$u_3 = ¼ [u_2 + 0 + 0 + 25]$

$4u_1 – u_2 = 25$ ------(3)

Solving equations (1), (2) and (3)

$$A = \begin{pmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{pmatrix}; X = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \text{ ; B =} \begin{pmatrix} 25 \\ 37.5 \\ 25 \end{pmatrix}$$

$$A^{-1} = 1/56 \begin{pmatrix} 15 & 4 & 1 \\ 4 & 16 & 4 \\ 1 & 4 & 15 \end{pmatrix}$$

$X = A^{-1} B$

We Get,
$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \frac{1}{56} \begin{pmatrix} 15 & 4 & 1 \\ 4 & 16 & 4 \\ 1 & 4 & 15 \end{pmatrix} \text{x} \begin{pmatrix} 25 \\ 37.5 \\ 25 \end{pmatrix}$$

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} 9.8214 \\ 14.2857 \\ 9.8214 \end{pmatrix}$$

$u_1 = 9.8214$; $u_2 = 14.2857$; $u_3 = 9.8214$

j \ i	**0**	**0.25**	**0.5**	**0.75**	**1**
0	0	18.75	25	18.75	0
Jan-16	0	9.8214	14.2857	9.8214	0

Coding:

```
clc;clear all;
deff('y=f(x,t)','y=100*(x-x^2)');
u=[f(0,0) f(0.25,0)f(0.5,0) f(0.75,0) f(1,0)];
disp("At (i=1,j=0)")
printf("\t\n4*u1-u2=25\n")
disp("At (i=2,j=0)")
printf("\t\n-u1+4*u2-u3=37.5\n")
disp("At (i=3,j=0)")
printf("\t\n-u2+4u3=25\n")
A=[4 -1 0;-1 4 -1;0 -1 4];
C=[25;37.5;25];
X=A^-1*C;
printf('\n\nu10=%f\t u20=%f\t u30=%f\t\n\n',u(2),u(3),u(4))
printf('\n\nu11=%f\t u21=%f\t u31=%f\t\n\n',X(1,1),X(2,1),X(3,1))
```

Output:

```
Scilab 6.1.1 Console
File  Edit  Control  Applications  ?
Scilab 6.1.1 Console
Warning : redefining function: f                       . Use funcprot(0) to avoid this message

  "At (i=1,j=0)"

4*u1-u2=25

  "At (i=2,j=0)"

-u1+4*u2-u3=37.5

  "At (i=3,j=0)"

-u2+4u3=25


u10=18.750000    u20=25.000000    u30=18.750000



u11=9.821429     u21=14.285714    u31=9.821429


-->
```

Exercise: 30

EXPLICIT SCHEME METHOD

Objective:

To find the solution for the given Partial Differential Equation using Explicit Scheme Method.

Input:

$$\frac{\partial^2 u}{\partial t^2} = 4\frac{\partial^2 u}{\partial x^2}$$

Boundary Conditions :

u (0, t) = u (4, t) = 0

u (x, 0) = x (4 – x)

Procedure/Methodology:

Solve the wave equation $u_{tt} = 4u_{xx}$ under conditions u (0, t) = u(4, t) = 0 and u (x, 0) = x (4 – x) taking h = 1.

Solution:

Standard form of wave equation in $u_{tt} = c^2 u_{xx}$

$c^2 = 4 \Rightarrow c = \pm 2$

Step length in x, h = 1
Step length in t, k = h/c = ½ = 0.5

By Boundary Condition
At x = 0,1,2,3,4 = i
At t = 1,0.5,1,1.5,2 = j

u(o, t) = u (4, t) = 0
u(x,0) = x(4 – x)
u(1,0) = 1(4 – 1) = 3
u(2,0) = 2(4 – 2) = 4
u(3,0) = 3(4 – 3) = 3

For j = 0

$u_{i,1} = ½ [u_{i+1,0} + u_{i-1,0}]$ r
$u_{1,1} = ½ [u_{2,0} + u_{0,0}] = 2$
$u_{2,1} = ½ [u_{3,0} + u_{1,0}] = 3$
$u_{3,1} = ½ [u_{4,0} + u_{2,0}] = 2$

General Formula

$u_{(i,j+1)} = u_{(i+1,j)} + u_{(i-1,j)} - u_{(i,j-1)}$

For j = 1

$u_{i,2} = u_{(i+1,1)} + u_{(i-1,1)} - u_{(i,0)}$

$u_{1,2} = u_{(2,1)} + u_{(0,1)} - u_{(1,0)} \quad = 0$

$u_{2,2} = u_{(3,1)} + u_{(1,1)} - u_{(2,0)} \quad = 0$

$u_{3,2} = u_{(4,1)} + u_{(2,1)} - u_{(3,0)} \quad = 0$

For j = 2

$u_{i,3} = u_{(i+1,2)} + u_{(i-1,2)} - u_{(i,1)}$

$u_{1,3} = u_{(2,2)} + u_{(0,2)} - u_{(1,1)} \quad = -2$

$u_{2,3} = u_{(3,2)} + u_{(1,2)} - u_{(2,1)} \quad = -3$

$u_{3,3} = u_{(4,2)} + u_{(2,2)} - u_{(3,1)} \quad = -2$

For j = 3

$u_{i,4} = u_{(i+1,3)} + u_{(i-1,3)} - u_{(i,1)}$

$u_{1,4} = u_{(2,3)} + u_{(0,3)} - u_{(1,2)} \quad = 0$

$u_{2,4} = u_{(3,3)} + u_{(1,3)} - u_{(2,2)} \quad = 0$

$u_{3,4} = u_{(4,3)} + u_{(2,3)} - u_{(3,2)} \quad = 0$

j \ i	**0**	**1**	**2**	**3**	**4**
0	0	3	4	3	0
0.5	0	2	3	2	0
1	0	0	0	0	0
1.5	0	-2	-3	-2	0
2	0	-3	-4	-3	0

Coding:

```
clc;
clear all;
```

```
deff('y=f(x)','y=4*x-x^2')
u=[f(0) f(1) f(2) f(3) f(4)];
b=0
printf('u01=%0.2f\t u02=%0.2f\t u03=%0.2f\t \n',f(1),f(2),f(3) )
u11=(u(1)+u(3))/2;
u12=(u(2)+u(4))/2;
u13=(u(3)+u(5))/2;
printf('u11=%0.2f\t u12=%0.2f\t u13=%0.2f\t \n',u11,u12,u13)
u21=b+u12-u(2);
u22=u11+u13-u(3);
u23=u12+b-u(4);
printf('u21=%0.2f\t u22=%0.2f\t u23=%0.2f\t \n',u21,u22,u23)
u31=b+u22-u11;
u32=u21+u21-u12;
u33=b+u22-u13;
printf('u31=%0.2f\t u32=%0.2f\t u33=%0.2f\t \n',u31,u32,u33)
u41=b+u32-u21;
u42=u33+u31-u22;
u43=b+u32-u23;
printf('u41=%0.2f\t u42=%0.2f\t u43=%0.2f\t \n',u41,u42,u43)
```

Output: