**RAJSHAHI UNIVERSITY OF ENGINEERING**

**& TECHNOLOGY**

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

COURSE N0: ECE - 2214

COURSE TITLE: NUMERICAL TECHNIQUES SESSIONAL

|  |  |
| --- | --- |
| **SUBMITTED BY:**  **NAME:MD SHAHARIAR HASAN RONOK**  **ROLL: 1710046**  **CLASS: 2nd YEAR, EVEN SEMESTER**  **SESSION: 2017-2018**  **DATE OF SUBMISSION: 07.10.2020** | **SUBMITTED TO:**  **NAME: Prof. Dr. Md. Shamim Anower**  **DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING**  **RAJSHAHI UNIVERSITY OF ENGINEERING & TECHNOLOGY**  **(RUET)** |

**ACKNOWLADGEMENT**

***THANK YOU SO MUCH MATHWORK FOR***

***GIVING US THE PERMISSION TO WORK IN***

***YOUR WORLD KNOWN SOFTWARE “MATLAB”.***

***WE ARE VERY MUCH GREATFUL TO YOU.***

**Experiment No: 01**

**Name of the Experiment:** Study of Bisection Method to Obtain the Roots of a Nonlinear Equation.

Bisection Method in MATLAB Code:

The given function is f(x) =2x^2-15x+3

y= @(x) 2\*x^2-15\*x+3 ;

while(1)

a=input('Enter the value of 1st assumption:');

b=input('Enter the value of 2nd assumption:');

if y(a)\*y(b)>0

fprintf('WRONG!!\n');

elseif y(a)\*y(b)<0

break;

end

end

if y(a)==0

fprintf('Root')

return

elseif y(b)==0;

fprintf('Root')

return

end

display('No. a b c y')

display('------------------------------------------')

a\_arr = [];

b\_arr = [];

c\_arr = [];

y\_arr = [];

i\_arr = [];

col={'a','b','c','y'};

for i=1:1:100

c=(a+b)/2;

if abs(y(c))<.001

break;

end

c\_arr(i) = c;

y\_arr(i) = y(c);

i\_arr(i)=i;

if y(a)\*y(c)>0

a=c;

a\_arr(i) = c;

if i == 1

b\_arr(i) = b;

else

b\_arr(i) = b\_arr(i-1);

end

else

b=c;

b\_arr(i) = c;

if i == 1

a\_arr(i) = a;

else

a\_arr(i) = a\_arr(i-1);

end

end

fprintf('%d %f %f %f %f \n',i,a,b,c,y(c));

uitable('columnname',col,'rowname',i\_arr,'data',[ a\_arr',b\_arr',c\_arr',y\_arr'],'position', [500 200 335 238] );%x,y,table decrease from right to left,

end

datatable = table(a\_arr', b\_arr', c\_arr', y\_arr','VariableNames',{'a','b','c','y'});

Output:

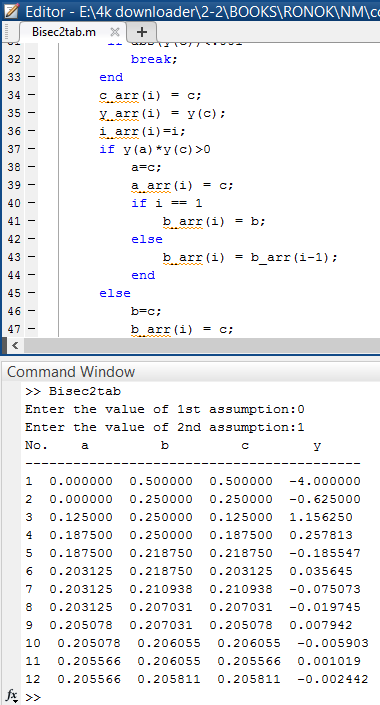
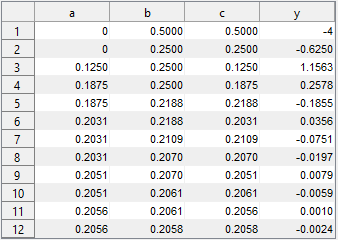
 

Fig1: Bisection method creating Table

**Results & Discussion:**

The resultant root of the given function is 0.2058.

**Experiment No: 02**

**Name of the Experiment:** Study of False Position Method to Obtain the Root(s) of a Nonlinear Equation.

MATLAB Code:

The given function is f(x) =2x^2-15x+3

y= @(x) 2\*x^2-15\*x+3 ;

while(1)

a=input('Enter the value of 1st assumption:');

b=input('Enter the value of 2nd assumption:');

if y(a)\*y(b)>0

fprintf('WRONG!!\n');

elseif y(a)\*y(b)<0

break;

end

end

if y(a)==0

fprintf('Root')

return

elseif y(b)==0;

fprintf('Root')

return

end

display(' No. a b c y')

display('---- ----- ----- ----- -----')

for i=1:1:100

c=(a\*y(b)-b\*y(a))/(y(b)-y(a));

if y(a)\*y(c)>0

a=c;

else b=c;

end

if abs(y(c))<.0001

break;

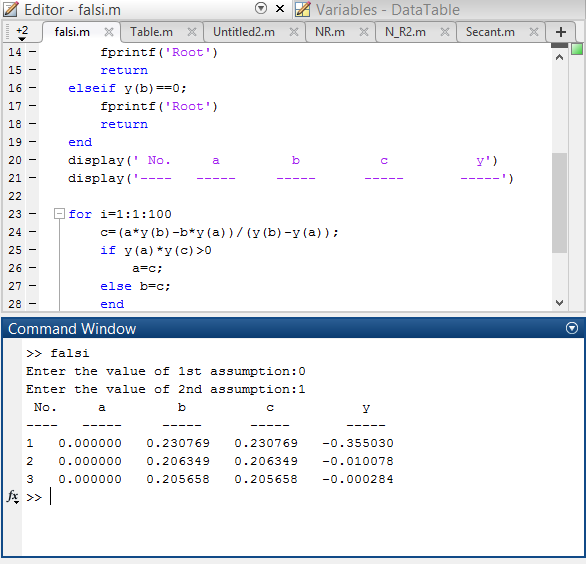
end

fprintf('%d %f %f %f %f \n',i,a,b,c,y(c));

datatables=table(a,b,c,y(c));

end

**Output:**

****

**Result& Discussion:** The roots of the given function is 0.205688.Which is nearly close to the original value (0.205638) direct calculated by calculator.

**Experiment No: 03**

**Name of the Experiment:** Study of Newton-Raphson(NR) Iterative Method to Obtain the Root(s) of a Nonlinear Equation.

MATLAB Code:

The given function is f(x) =2x^2-15x+3

clear all

clc

syms x;

fun=input('Enter the fun:');

f=inline(fun);

z=diff(f(x));

f1=inline(z);

x0=input('Enter initial value:');

x=x0;

display(' No. y f(a) f1(a) x')

display('---- ----- ----- ----- -----')

for i=0:1:15

y=x;

x=y-(f(x)/f1(x));

if x==y

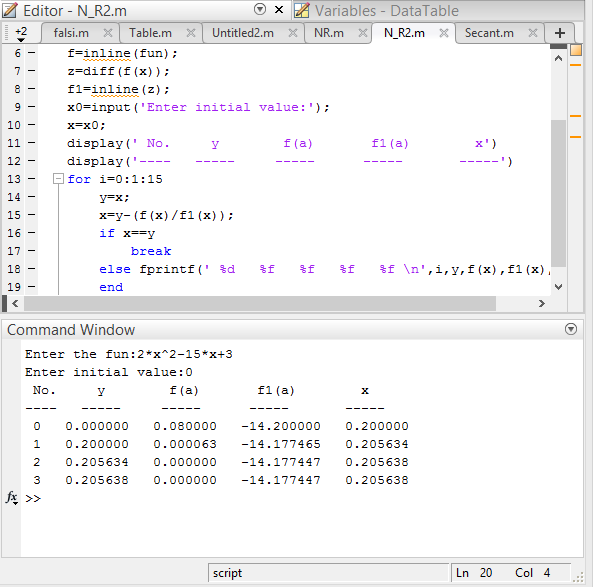
break

else fprintf(' %d %f %f %f %f \n',i,y,f(x),f1(x),x);

end

end

**Output:**

****

**Result& Discussion:** The roots of the given function is 0.205638.Which is equal to the original value (0.205638) directly calculated by calculator.

**Experiment No: 04**

**Name of the Experiment:** Study of Secant Method to Obtain the Root(s) of a Nonlinear Equation.

**MATLAB Code:**

The given function is f(x) =2x^2-15x+3

clear all

clc

syms x;

fun=input('Enter the fun:');

f=inline(fun);

while(1)

a=input('Enter the value of 1st assumption:');

b=input('Enter the value of 2nd assumption:');

if f(a)\*f(b)>0

disp('Wrong');

elseif f(a)\*f(b)<=0

break;

end

end

if f(a)==0

fprintf('Root')

return

elseif f(b)==0;

fprintf('Root')

return

end

display(' No. a b xn ')

display('---- ----- ----- ----- ')

for i=1:1:100

x=a-b;

z=f(a)-f(b);

xn=a-(x/z)\*f(a);

if xn==a

break

else fprintf(' %d %f %f %f\n',i,a,b,xn);

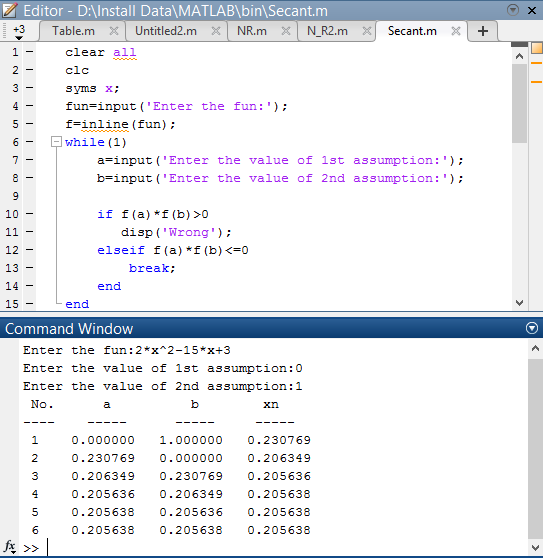
end

b=a;

a=xn;

end

**Output:**

****

**Result& Discussion:** The roots of the given function is 0.205638.Which is equal to the original value (0.205638) directly calculated by calculator.

**Experiment No: 05**

**Name of the Experiment:** Study of Successive Approximations (SA) Method to Obtain the Root(s) of a Nonlinear Equation.

MATLAB Code:

The given function is f(x) =(x^3+3)/5

clear all

clc

syms x;

fun=input('Enter the fun:');

f=inline(fun);

a=input('Enter the value of initial assumption:');

if f(a)==0

fprintf('Root')

return

end

display(' No. a xn ')

display(' -- ----- ----- ')

for i=1:1:20

xn=f(a);

if abs(xn-a)<0.001

break;

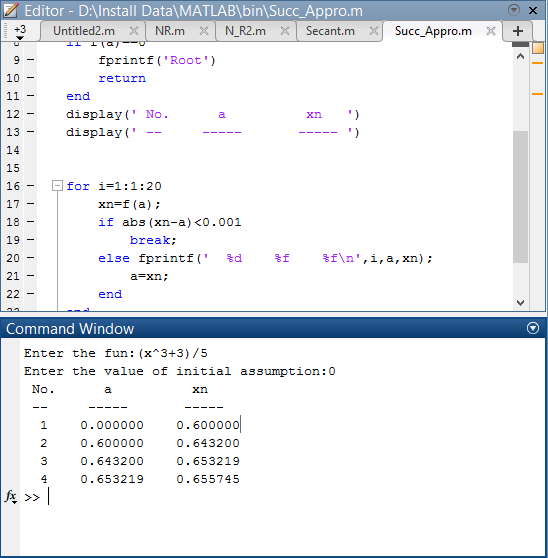
else fprintf(' %d %f %f\n',i,a,xn);

a=xn;

end

end

**Output:**

****

**Result& Discussion:** The roots of the given function is 0.655745.Which is nearly close to the original value (0.65634) direct calculated by calculator.

**Experiment No: 06**

**Name of the Experiment:** Study of Gauss Elimination (GE) Method to Find the Solution of Simultaneous Equations.

MATLAB Code:

A=[1 1 1; 2 1 3; 3 4 -2];

B=[4;7;9];

% Augmented matrix

AB=[A,B];

%% pivot 1 1

alpha = AB(2,1)/AB(1,1);

AB(2,:)=AB(2,:)-alpha\*AB(1,:);

alpha=AB(3,1)/AB(1,1);

AB(3,:)=AB(3,:)-alpha\*AB(1,:);

%% pivot 2 2

alpha=AB(3,2)/AB(2,2);

AB(3,:)=AB(3,:)-alpha\*AB(2,:);

%% Back Subs

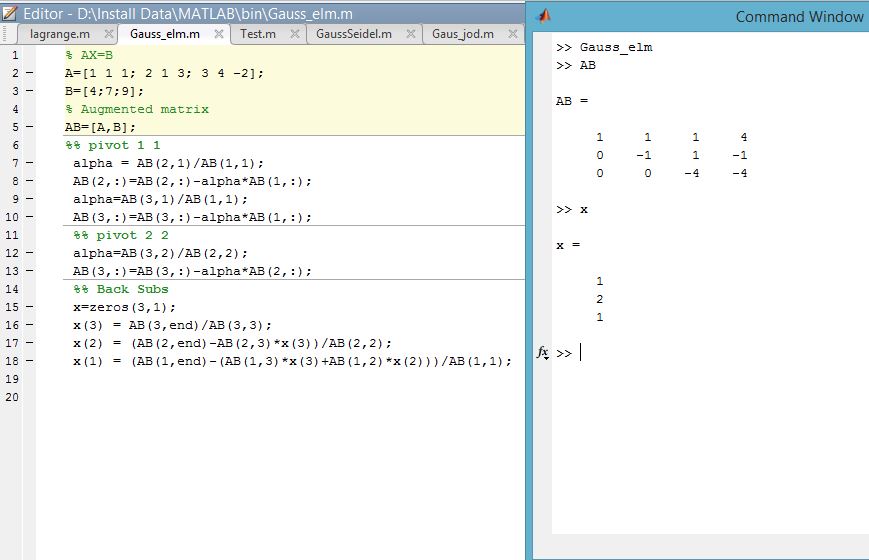
x=zeros(3,1);

x(3) = AB(3,end)/AB(3,3);

x(2) = (AB(2,end)-AB(2,3)\*x(3))/AB(2,2);

x(1) = (AB(1,end)-(AB(1,3)\*x(3)+AB(1,2)\*x(2)))/AB(1,1);

**Output:**

****

**Result& Discussion:** From the output the value of x matrix is 1, 2, 1.Means x=1, y=2, z=1.

**Experiment No: 07**

**Name of the Experiment:** Study of Gauss Jordan (GJ) Method to Find the Solution of Simultaneous Equations.

**MATLAB Code:**

A=[1 1 1; 2 1 3; 3 4 -2];

B=[4;7;9];

% Augmented matrix

AB=[A,B];

%% pivot 1 1

alpha = AB(2,1)/AB(1,1);

AB(2,:)=AB(2,:)-alpha\*AB(1,:);

alpha=AB(3,1)/AB(1,1);

AB(3,:)=AB(3,:)-alpha\*AB(1,:);

%% pivot 2 2

alpha=AB(1,2)/AB(2,2);

AB(1,:)=AB(1,:)-alpha\*AB(2,:);

alpha=AB(3,2)/AB(2,2);

AB(3,:)=AB(3,:)-alpha\*AB(2,:);

%% pivot 3 3

alpha=AB(1,3)/AB(3,3);

AB(1,:)=AB(1,:)-alpha\*AB(3,:);

alpha=AB(2,3)/AB(3,3);

AB(2,:)=AB(2,:)-alpha\*AB(3,:);

%% Back Subs

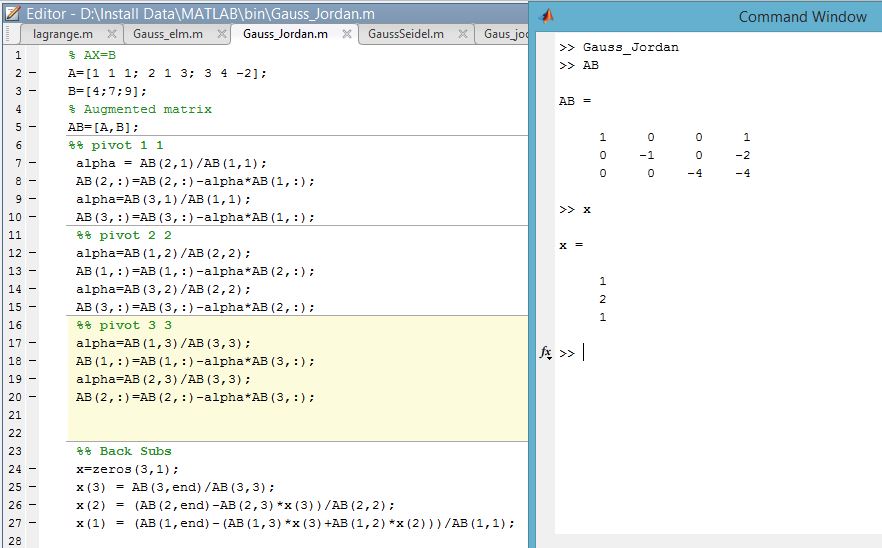
x=zeros(3,1);

x(3) = AB(3,end)/AB(3,3);

x(2) = (AB(2,end)-AB(2,3)\*x(3))/AB(2,2);

x(1) = (AB(1,end)-(AB(1,3)\*x(3)+AB(1,2)\*x(2)))/AB(1,1);

**Output:**

****

**Result& Discussion:** From the output the value of x matrix is 1, 2, 1.Means x=1, y=2, z=1.

**Experiment No: 08**

**Name of the Experiment:** Study of Gauss Seidel (GS) Method to Find the Solution of Simultaneous Equations.

**MATLAB Code:**

f1 =@(x,y,z) (1/20)\*(17-y+2\*z);

f2 =@(x,y,z) (1/20)\*(-18-3\*x+z);

f3 =@(x,y,z) (1/20)\*(25-2\*x+3\*z);

a\_arr = [];

b\_arr = [];

c\_arr = [];

x=0;y=0;z=0;

for i=1:20

a=x;

x=f1(x,y,z);

if(abs(x-a)<0.001)

break;

end

y=f2(x,y,z);

z=f3(x,y,z);

a\_arr(i) = x;

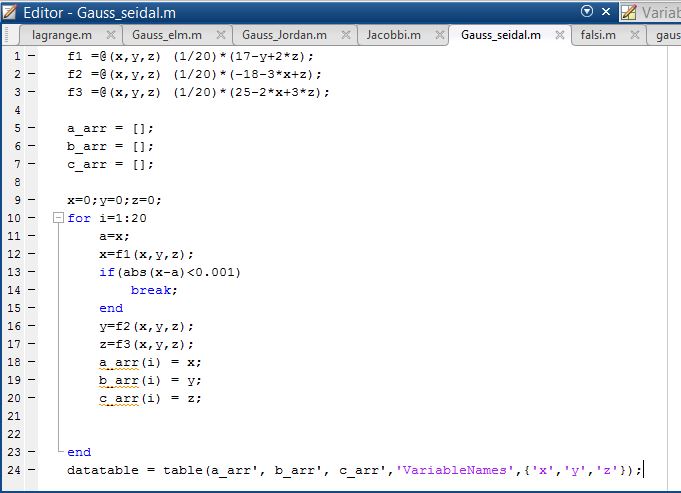
b\_arr(i) = y;

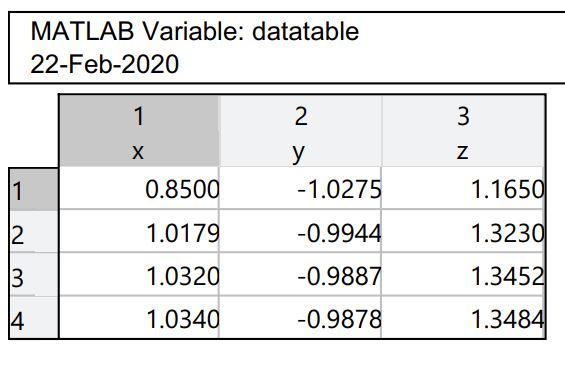
c\_arr(i) = z;

end

datatable = table(a\_arr', b\_arr', c\_arr','VariableNames',{'x','y','z'});

**Output:**

****

****

**Figure 09: Data Table**

**Result& Discussion:** The values of the given function is are 1.0430,-9878, 1.3484.Which is nearly close to the original values (1.043,-988, 1.34) direct calculated by calculator.

**Experiment No: 09**

**Name of the Experiment:** Study of Jacobi Method to Find the Solution of Simultaneous Equations.

**MATLAB Code:**

%declaring functions

f1 =@(x,y,z) (1/20)\*(17-y+2\*z);

f2 =@(x,y,z) (1/20)\*(-18-3\*x+z);

f3 =@(x,y,z) (1/20)\*(25-2\*x+3\*z);

a\_arr = [];

b\_arr = [];

c\_arr = [];

x=0;y=0;z=0;

%Iteration

for i=1:20

a=f1(x,y,z);

b=f2(x,y,z);

c=f3(x,y,z);

if(abs(x-a)<0.001)

break;

end

a\_arr(i) = a;

b\_arr(i) = b;

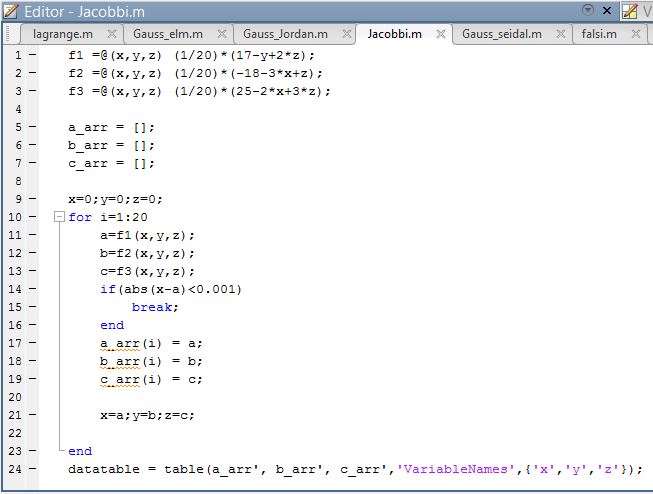
c\_arr(i) = c;

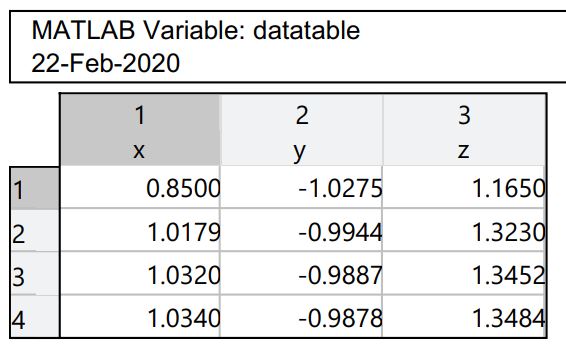
x=a;y=b;z=c;

end

datatable = table(a\_arr', b\_arr', c\_arr','VariableNames',{'x','y','z'});

**Output:**

****

****

**Figure 10: Data Table**

**Result& Discussion:** The values of the given function is are 1.0430,-9878, 1.3484.Which is nearly close to the original values (1.043,-988, 1.34) direct calculated by calculator.

**Experiment No: 10**

**Name of the Experiment:** Study of Lagrange Interpolation Method to Predict the Unknown Value(s) For Any Geographic Point Data.

**MATLAB Code:**

x = [1.5 3 6];

y = [-.25 2 20];

n = size(x, 2);

%value for fining fx

x\_int = 4;

%% determining sum by formula

y\_int = 0;

for i = 1:n

p = y(i);

for j = 1:n

if i ~= j

p = p \* ((x\_int - x(j)) / (x(i) - x(j)));

end

end

y\_int = y\_int + p;

end

y\_int

%% ploting the graph

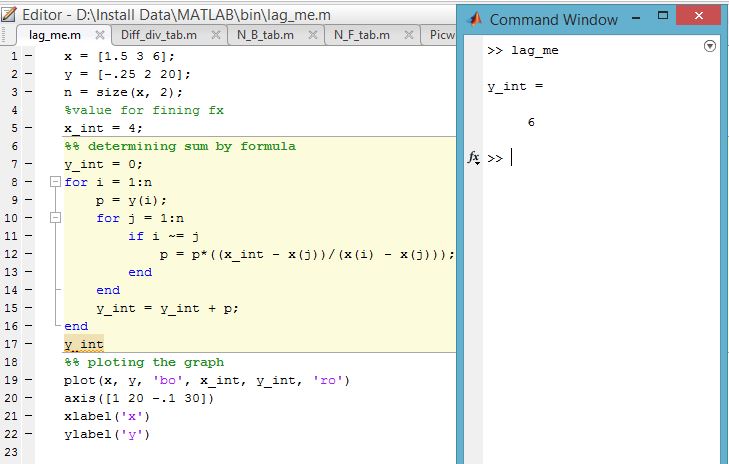
plot(x, y, 'bo', x\_int, y\_int, 'ro')

axis([1 20 -.1 30])

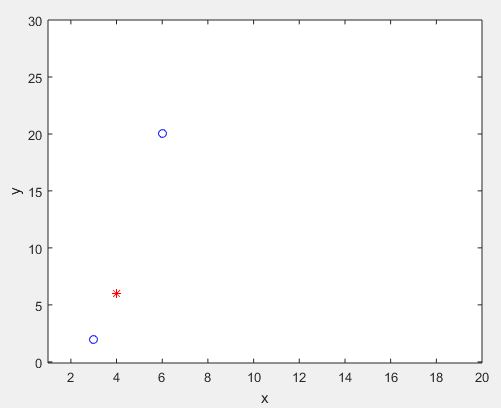
xlabel('x')

ylabel('y')

**Output:**

****

**---------------------------------------------------------**

****

**Figure 11.1: Graph Of The Function**

**Result(s)& Discussion:** The unknown value for x = 4 is y = 6.From text book for x=4 is y=5.99 ~ 6

**Experiment No: 11**

**Name of the Experiment:** Study of Divided Difference Method to Predict Unknown Value(s) For Any Geographic Point Data.

**MATLAB CODE:**

x=[-3 -1 0 3 5];

fx=[-30 -22 -12 330 3458];

n=size(x,2);

%array of zeros

dt=zeros(n+1,n+1);

%% inserting x and fx in dt

for i=1:n

dt(i,1)=x(i);

dt(i,2)=fx(i);

end

%% creating divided difference table

z=3;l=0;k=2;

for i=1:n-1

for j=k:n

dt(j,z)=(dt(i+1,z-1)-dt(i,z-1))/(dt(i+1,1)-dt(i-l,1));

i=i+1;

if(i>=n)

break;

end

end

k=k+1;l=l+1;z=z+1;

end

%value for fining fx

x\_int=2.5;

%% determining sum by formula

y\_sum=dt(1,2);

for i=2:n

d=1;

for j=1:i-1

d=d\*(x\_int - x(j));

end

y\_sum=y\_sum+dt(i,i+1)\*d;

end

%% result

dt

y\_sum

%% ploting the graph

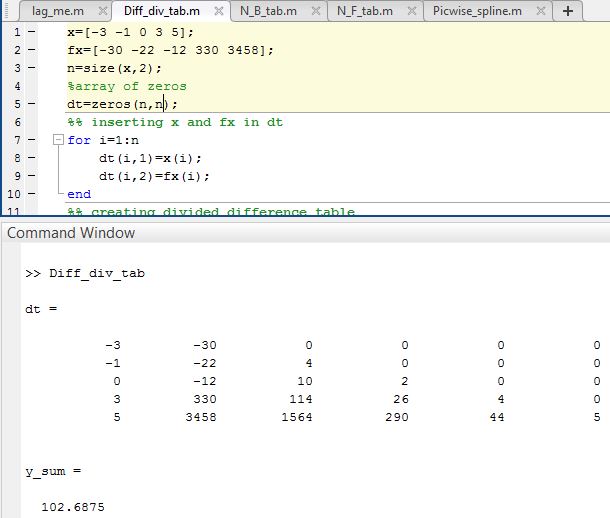
plot(x, fx, 'bo', x\_int, y\_sum, 'ro')

axis([-10 10 -1000 4000])

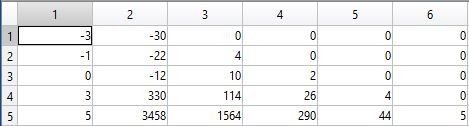
xlabel('x')

ylabel('y')

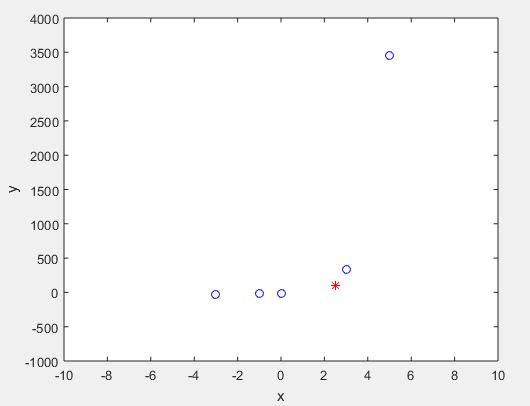
**Output:**

****

**------------------------------------------------------**

****

**Figure 12.1: Table of Newton Divided Difference**

****

**Figure 12.2: Graph Of The Function**

**Result(s)& Discussion:** The unknown values for x = 2.5 is y = 102.6875 . From text book[1] for x=2.5 is y=102.7

**Experiment No: 12**

**Name of the Experiment:** Study of Newton Forward Difference Method to Predict Unknown Value(s) for Any Geographic Point Data.

**MATLAB Code:**

x=[1921 1931 1941 1951 1961 1971];

fx=[35 42 58 84 120 165];

n=size(x,2);

%array of zeros

dt=zeros(n,n);

%% inserting x and fx in dt

for i=1:n

dt(i,1)=x(i);

dt(i,2)=fx(i);

end

dt

z=3;

for k=1:n-1

i=1;

for j=1:n-k

dt(j,z)=(dt(i+1,z-1)-dt(i,z-1));

i=i+1;

if(j>n)

break;

end

end

z=z+1;k=k+1;

end

dt

%value for fx to find

x\_int=1947;

%determining u=(x-x1)\*h

u=(x\_int-x(n/2))/(x(2)-x(1));

%% determining sum by formula

y\_sum=dt(n/2,2)+u\*(dt(n/2,3)+dt(n/2-1,3))/2+(u\*u)/2\*(dt(n/2-1,4));

a=0;b=0;l=1;t=1;

for i=3:n-1

k=1;

if((n/2-(i-2))<=0 || (n/2-(i-1))<=0)%cross limit dt(0,i)

break;

else

a=(dt(n/2-(i-2),i+2)+dt(n/2-(i-1),i+2))/2;

b=dt(n/2-(i-1),i+3);

if(i<=n/2)

l=u\*(u\*u-t\*t);

t=t+1;

end

for j=1:i

k=k\*j;

end

y\_sum=y\_sum+(l/k)\*a+u\*l\*b;

end

end

y\_sum

%% ploting the graph

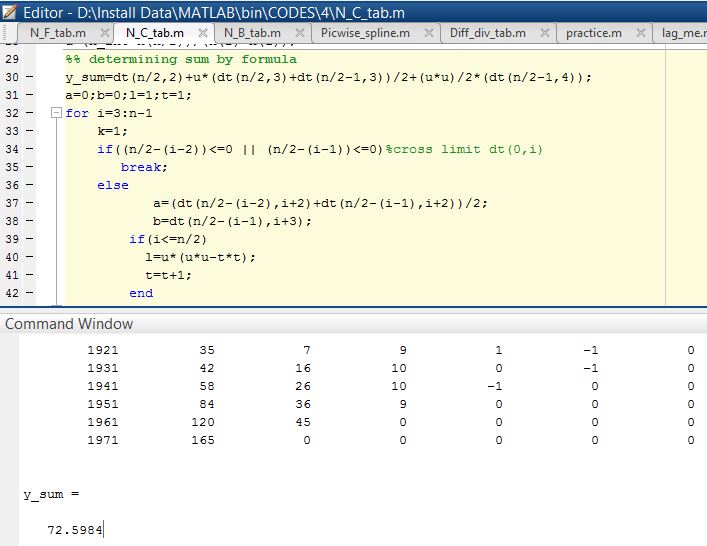
plot(x, fx, 'bo', x\_int, y\_sum, 'r\*')

axis([1900 2000 0 250])

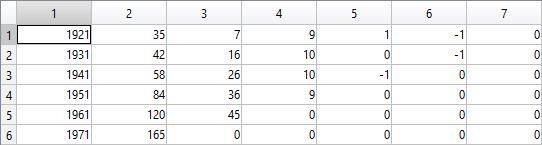
xlabel('x')

ylabel('y')

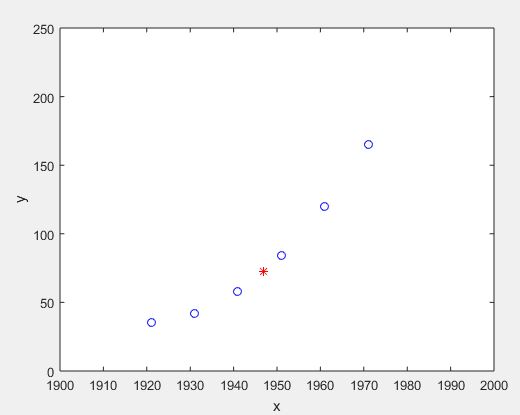
**Output:**

****

**-----------------------------------------------------------------------**

****

**Figure 13.1: Table of Newton Forward Difference**

****

**Figure 13.2: Graph Of The Function**

**Result(s) & Discussion:** The unknown values for x = 1947 is y = 72.5984.

**Experiment No: 13**

**Name of the Experiment:** Study of Newton Backward Difference Method to Predict Unknown Value(s) for Any Geographic Point Data.

**MATLAB Code:**

x=[1921 1931 1941 1951 1961 1971 1981];

fx=[35 42 58 84 120 165 220];

n=size(x,2)

%array of zeros

dt=zeros(n,n)

%% inserting x and fx in dt

for i=1:n

dt(i,1)=x(i);

dt(i,2)=fx(i);

end

%% creating Newton backward difference table

z=3;k=2;

for i=1:n-1

for j=k:n

dt(j,z)=(dt(i+1,z-1)-dt(i,z-1));

i=i+1;

if(i>=n)

break;

end

end

k=k+1;z=z+1;

end

%value for fx to find

x\_int=1975;

%determining u=(x-x1)\*h

u=(x\_int-x(n))/(x(2)-x(1));

%% determining sum by formula

y\_sum=dt(n,2);k=1;d=1;

for i=2:4

for j=0:i-2

d=d\*(u + j);

k=k\*(j+1);

end

y\_sum=y\_sum+(dt(n,i+1)/k)\*d;

d=1;

dt(n,i+1);

end

%% result

dt

y\_sum

%% ploting the graph

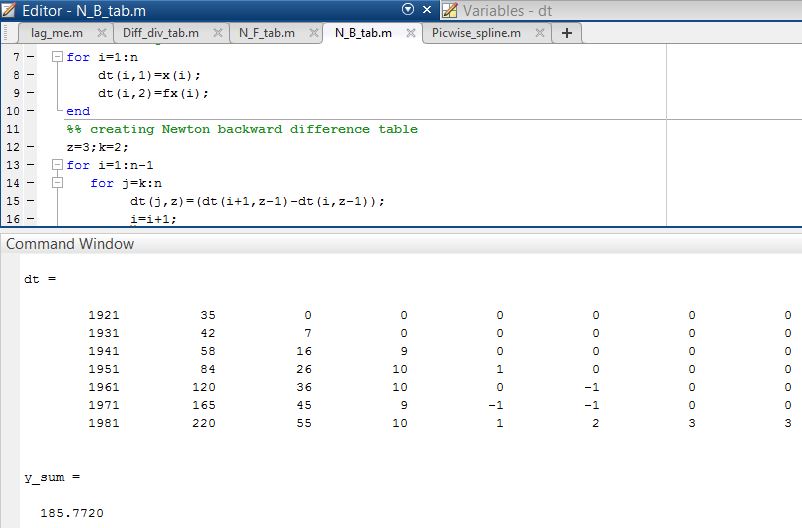
plot(x, fx, 'bo', x\_int, y\_sum, 'r\*')

axis([1900 2000 0 250])

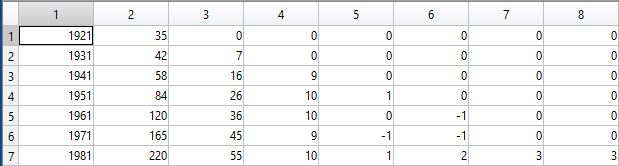
xlabel('x')

ylabel('y')

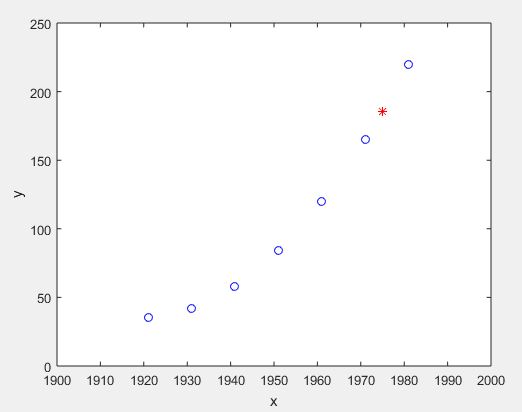
**Output:**

****

**------------------------------------------------------**

****

**Figure 14.1: Table of Newton Backward Difference**

****

**Figure 14.2: Graph Of The Function**

**Result(s)& Discussion:** The unknown values for x = 1975 is y = 185.7720 . From text book[1] for x=1975 is y=185.8=186(round)

**Experiment No: 14**

**Name of the Experiment:** Study Of Piecewise Linear Fit Interpolation Method To Predict Unknown Value(s) For Any Geographic Point Data.

**MATLAB Code:**

x = [1 2 3 4 5 6];

y = [33 16 35 25 35 26];

%value for fx to find

x\_int = 3.7;

%% using formula

for i=2:7

if(x\_int <= x(i))

i=i-1;

f=(y(i+1)\*(x\_int - x(i))-y(i)\*(x\_int-x(i+1)))/(x(i+1)-x(i));

break;

end

end

%% result

x\_int

f

%% ploting the graph

hold on

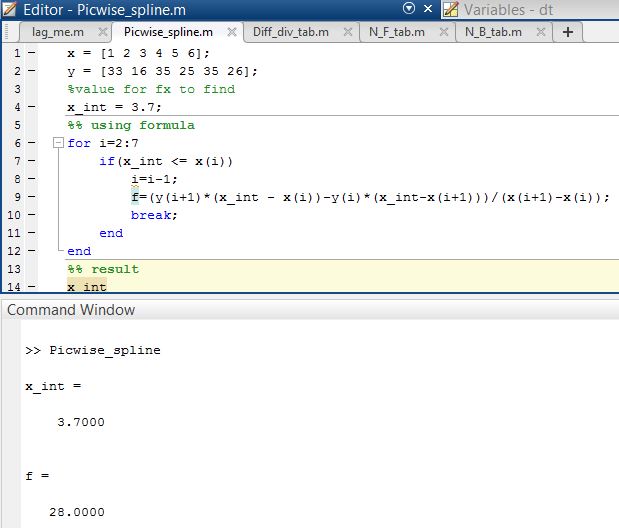
plot(x, y, x\_int, f,'ro')

axis([0 10 10 50])

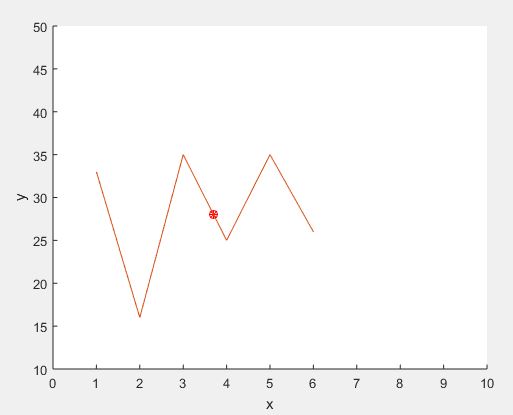
xlabel('x')

ylabel('y')

**Output:**

****

**--------------------------------------------------------**

****

**Figure 15.1: Graph of the Function**

**Result(s)& Discussion:** The unknown values for x = 3.7 is y = 28 . From text book[1] for x=3.7 is y=28

**Experiment No: 15**

**Name of the Experiment:** Study of Trapezoidal Integral Method to Calculate Integral Value of a Function with Limit.

**MATLAB Code:**

%Function declaration

f= @(x) 2\*x+3 ;

a=0;b=6;

n=b-a;

%height

h = (b - a) / n;

%Trapizoidal formula

sum\_x = 0;

for i = 1:(n - 1)

x = a + i \* h;

sum\_x = sum\_x + f(x);

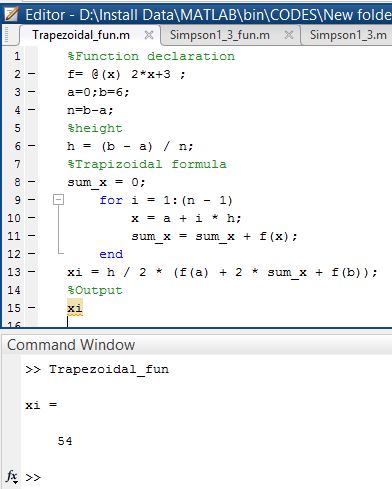
end

xi = h / 2 \* (f(a) + 2 \* sum\_x + f(b));

%Output

xi

**Output:**

****

**Result(s)& Discussion:** The integral value is 54.

**Experiment No: 16**

**Name of the Experiment:** Study of Simpson’s 1/3 Integral Method to Calculate Integral Value of a Function with Limit.

**MATLAB Code:**

%Function declaration

f= @(x) 3\*x^2+3;

a=0;b=6;

n=b-a;

%height

h = (b - a) / n;

%simpson's formula

sum\_x1 = 0;sum\_x2=0;

for i=0:1:n

x = a + i \* h;

if(i>1 && i<n && mod(i,2)==0)

sum\_x2=sum\_x2+f(x);

end

if (i>0 && i<n && mod(i,2)~=0)

sum\_x1 = sum\_x1 + f(x);

end

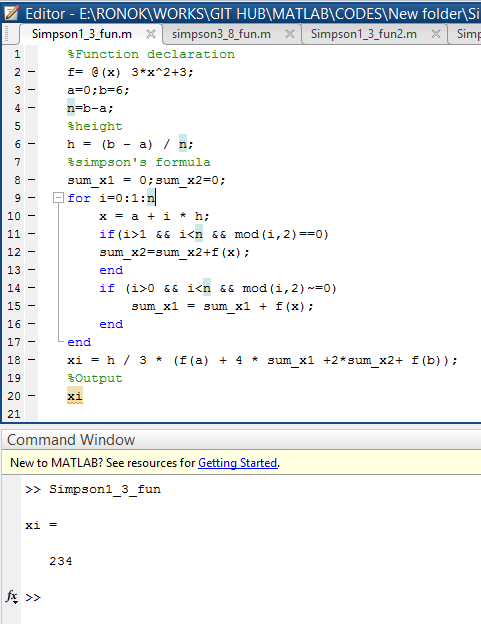
end

xi = h / 3 \* (f(a) + 4 \* sum\_x1 +2\*sum\_x2+ f(b));

%Output

xi

**Output:**

****

**Result(s)& Discussion:** The integral value is 234.

**Experiment No: 17**

**Name of the Experiment:** Study of Simpson’s 3/8 Integral Method to Calculate Integral Value of a Function with Limit.

**MATLAB Code:**

%Function declaration

f= @(x) 3\*x^2+3;

a=0;b=6;

n=b-a;

%height

h = (b - a) / n;

%simpson's formula

sum\_x1 = 0;sum\_x2=0;

for i=0:1:n

x = a + i \* h;

if(i>1 && i<n && mod(i,3)==0)

sum\_x2=sum\_x2+f(x);

end

if (i>0 && i<n && mod(i,3)~=0)

sum\_x1 = sum\_x1 + f(x);

end

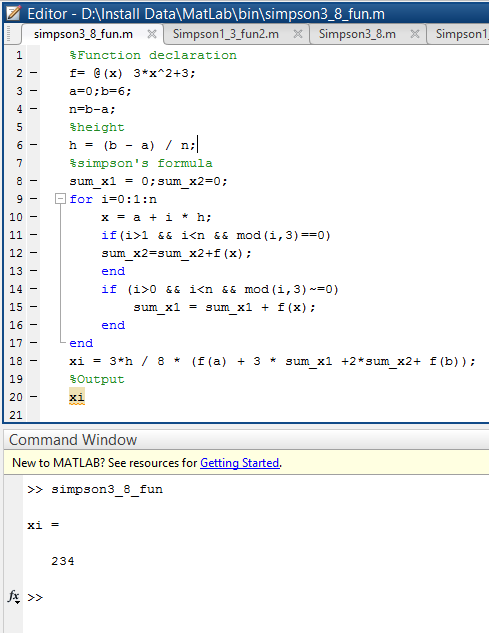
end

xi = 3\*h / 8 \* (f(a) + 3 \* sum\_x1 +2\*sum\_x2+ f(b));

%Output

xi

**Output:**

****

**Result(s)& Discussion:** The integration value is 234.

**Experiment No: 18**

**Name of the Experiment:** Study of Euler’s Method to Solve Ordinary Differential Equation(s) (Initial Value Problem)

**Problem:** The temperature radiation of a ball in air at ambient temperature 300K can be describe by the differential equation

= −𝟐. 𝟐𝟎𝟔𝟕 × 𝟏𝟎-𝟏𝟐(∅𝟒 − 𝟖𝟏 × 𝟏𝟎𝟖)

Using Euler’s method find the temperature of the ball at 𝒕 = 𝟒𝟖𝟎 seconds where ∅ is in K and t in second. It is assumed that the initial temperature of the ball is 1200K.

**MATLAB Code:**

clear all;

close all;

clc;

f=@(x,y)-2.2067\*10^-12\*(y^4-81\*10^8);

h=60;

n=7;

y0=1200;

x0=0;

xi=x0;

yi=y0;

for i=0:n

yi=yi+h\*f(xi,yi);

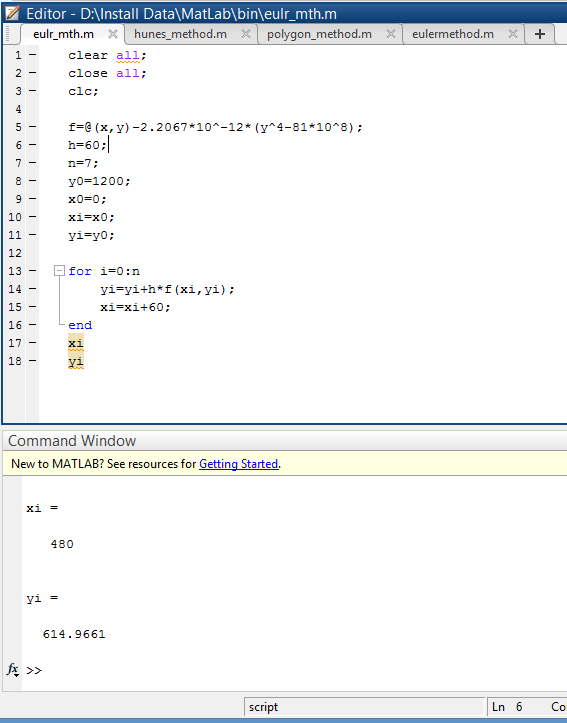
xi=xi+60;

end

xi

yi

**Output:**

****

**Result(s)& Discussion:** The result is ∅(480) = ∅8= 614.9661K

**References:** PDF provided by **Prof. Dr. Md. Shamim Anower**

**Experiment No: 19**

**Name of the Experiment:** Study of Heun’s Method to Solve Ordinary Differential Equation(s) (Initial Value Problem)

**Problem:** The temperature radiation of a ball in air at ambient temperature 300K can be describe by the differential equation

= −𝟐. 𝟐𝟎𝟔𝟕 × 𝟏𝟎-𝟏𝟐(∅𝟒 − 𝟖𝟏 × 𝟏𝟎𝟖)

Using Euler’s method find the temperature of the ball at 𝒕 = 𝟒𝟖𝟎 seconds where ∅ is in K and t in second. It is assumed that the initial temperature of the ball is 1200K.

**MATLAB Code:**

clear all;

close all;

clc;

f=@(x,y)-2.2067\*10^-12\*(y^4-81\*10^8);

h=60;

n=7;

y0=1200;

x0=0;

xi=x0;

yi=y0;

for i=0:n

s1=f(xi,yi);

s2=f(xi+h,yi+s1\*h);

yi=yi+h/2\*(s1+s2);

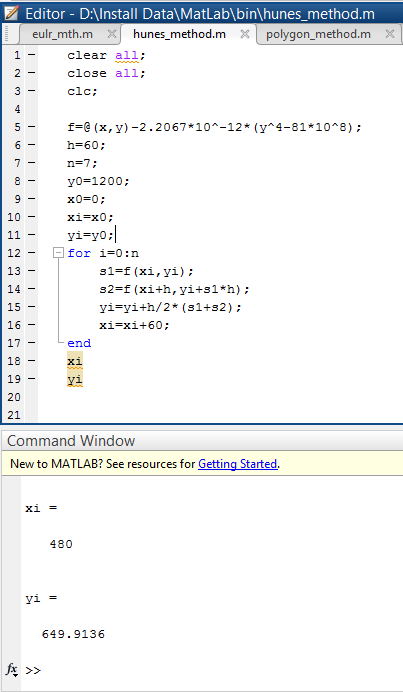
xi=xi+60;

end

xi

yi

**Output:**

****

**Result(s)& Discussion:** The result is ∅(480) = ∅8= 649.9136K

**Experiment No: 20**

**Name of the Experiment:** Study of Polygon Method to Solve Ordinary Differential Equation(s) (Initial Value Problem)

**Problem:** The temperature radiation of a ball in air at ambient temperature 300K can be describe by the differential equation

= −𝟐. 𝟐𝟎𝟔𝟕 × 𝟏𝟎-𝟏𝟐(∅𝟒 − 𝟖𝟏 × 𝟏𝟎𝟖)

Using Euler’s method find the temperature of the ball at 𝒕 = 𝟒𝟖𝟎 seconds where ∅ is in K and t in second. It is assumed that the initial temperature of the ball is 1200K.

**MATLAB Code:**

clear all;

close all;

clc;

f=@(x,y)-2.2067\*10^-12\*(y^4-81\*10^8);

h=60;

n=7;

y0=1200;

x0=0;

xi=x0;

yi=y0;

for i=0:n

si=f(xi,yi);

yi=yi+h\*f(xi+h/2,yi+si\*(h/2));

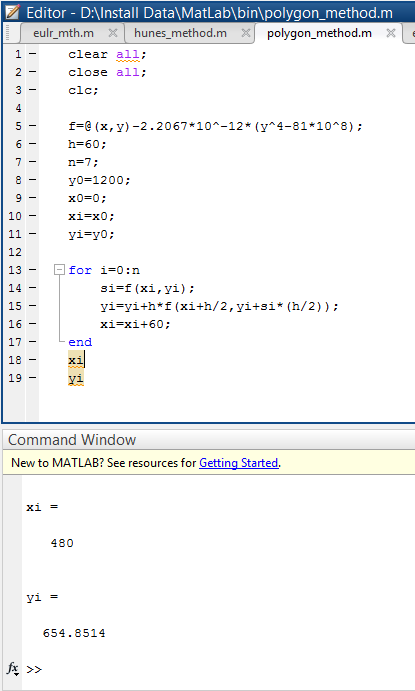
xi=xi+60;

end

xi

yi

**Output:**

****

**Result(s) & Discussion:** The result is ∅ (480) = ∅8= 654.8514K