**EXPERIMENT 3**

**INTERPOLATION METHODS**

**1.Direct Method**

Ques:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| T(s) | 0 | 10 | 15 | 20 | 22.5 |
| V(m/s) | 0 | 227.04 | 362.78 | 517.35 | 602.97 |

**CODE**

clc

clear all

n=input('interpolation polynomial order: '); %1

x=input('polynomial node values in ascending order: '); %[15;20]

y=input('enter y values: ');%[362.78;517.35]

xg=input('Enter value of xg at which y is calculated: ');

yf=0;

A=zeros(n+1);

B=zeros(n+1,1);

C=zeros(n+1,1);

for i=1:n+1

for j=1:n+1

A(i,j)=(x(i,1)^(j-1));

end

B=y;

end

A

C=inv(A)\*B

for i=1:n+1

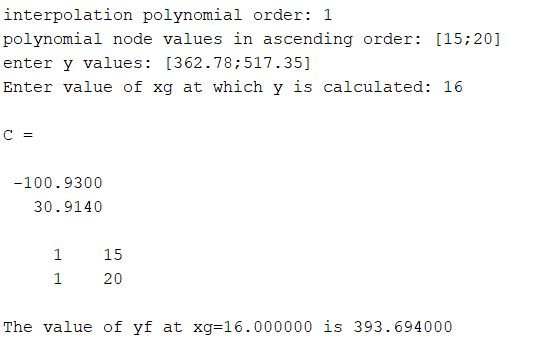
yf=yf+C(i,1)\*xg^(i-1);

end

fprintf('The value of yf at xg=%f is %f\n',xg,yf);

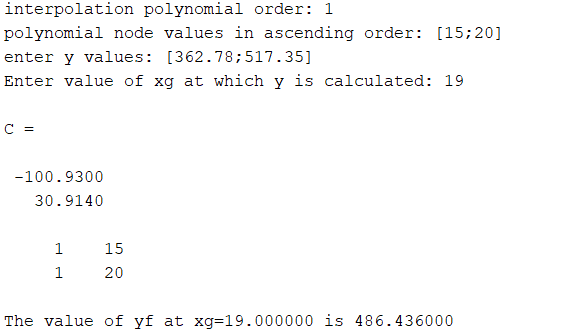
**CASE 1: T=16s**

**OUTPUT**

****

**CASE 2: T=19s**

**OUTPUT**

****

**2.Lagrange’s Method**

Ques:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x(ohms) | 1101 | 911.3 | 636 | 451.1 |
| y(sec) | 25.113 | 30.131 | 40.120 | 50.128 |

**CODE**

clc

clear all

x=[1101 911.3 636 451.1];

y=[25.113 30.131 40.120 50.128];

xg=input('enter the value of xg: ');

for i=1:length(x)

xc=x; %xc copy of x

xc(i)=[]; %removing ith node point from x

p(i)= prod(xg-xc)/prod(x(i)-xc);

end

y1=sum(p.\*y);

fprintf('value of y at xg=%f is %f\n',xg,y1);

**OUTPUT**

****

**3. Newton’s Divided Difference Method**

Ques:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | 4 | 7 | 9 | 12 |
| y | -43 | 83 | 327 | 1053 |

**CODE**

clc

clear all

n=input('enter the number of datapoint= ');

Y=zeros(n,n);

for i=1:n

X(i)=input('enter the value of x='); %TAKE VALUE IN ROW

Y(i,1)=input('enter the value of y=');

end

k=0;

for i=2:n

for j=1:n-i+1 %skip the last one in matrix

num=(Y(j+1,i-1)-Y(j,i-1));

den=(X(j+1+k)-X(j));

Y(j,i)=num/den;

end

k=k+1;

end

x=input('enter the value of x=');

y=0;

r=1;

for i= 1:n

if i>1

r=r\*(x-X(i-1));

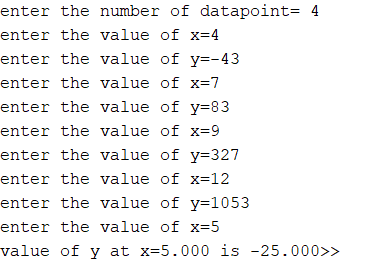
end

y=y+(Y(1,i)\*(r));

end

fprintf('value of y at x=%0.3f is %0.3f',x,y);

**OUTPUT**

****

**4.Newton’s Forward Interpolating Method**

Ques:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | 1 | 3 | 5 | 7 |
| y | 24 | 120 | 336 | 720 |

**CODE**

clc

clear all

n = input("enter value: ");

for a=1:n

X(a)=input("enter the value of array x");

end

for b=1:n

Y(b)=input("enter the value of array y");

end

x= input("enter the value of t:");

h = X(2)-X(1);

for i= 1:n-1

d(i,1)=Y(i+1)-Y(i);

end

for j=2:n-1

for i=1:n-j

d(i,j)=d(i+1,j-1)-d(i,j-1);

end

end

d

p=(x-X(1))/h;

prod=1;

y=Y(1);

for t=1:n-1

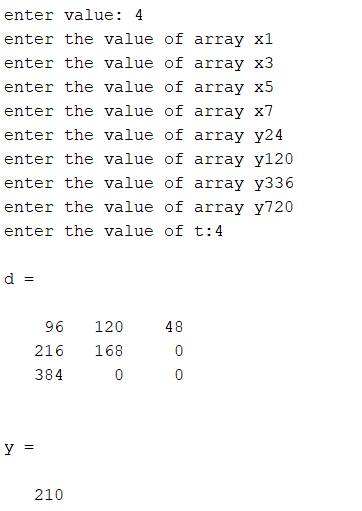
prod=prod\*(p-t+1)/t;

y=y+prod\*d(1,t);

end

y

**OUTPUT**

****

**5.Newton’s Backward Method**

Ques:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | 1891 | 1901 | 1911 | 1921 | 1931 |
| y | 46 | 66 | 81 | 93 | 101 |

**CODE**

clc

clear all

n = input("enter value: ");

for a=1:n

X(a)=input("enter the value of array x: ");

end

for b=1:n

Y(b)=input("enter the value of array y: ");

end

x= input("enter the value of t:");

h = X(2)-X(1);

for i= 1:n-1

d(i,1)=Y(i+1)-Y(i)

end

for j=2:n-1

for i=1:n-j

d(i,j)=d(i+1,j-1)-d(i,j-1);

end

end

d

p=(x-X(n))/h;

prod=1;

y=Y(n);

r=n-1;

for t=1:n-1

prod=prod\*(p+t-1)/t;

y=y+prod\*d(r,t);

r=r-1;

end

y

**OUTPUT**

