**Newton’s Formula for Interpolation**

**Introduction:** Interpolation is the process of approximating a given function, whose values are known at n+1 tabular points, by a suitable polynomial, Yn(x)of degree n which takes the values yi at x=xi for i=0,1,2………n

In the following, we shall use forward and backward differences to obtain polynomial function approximating y=f(x) when the tabular points xi's are equally spaced.i.e. let

xi=x0+ph, p=0,1,2,……,n here, h=difference.

=>P=( xi-x0)/h

***Newton Forward Difference formula of Interpolation:***

Yn(x)=y0+ pΔy0

Let find the polynomial

Y(1)=24 Y(3)=120

Y(5)=336 Y(7)=720

Y(8)=?

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | x | Y | Δy0 | Δ2y0 | Δ3y0 | | 1 | 24 | 96 |  |  | | 3 | 120 | 120 | 48 | | 216 | | 5 | 336 | 168 |  | | 384 | | 7 | 720 |  |  | |  | | 8 | ? |  |  | |

h=2

so, p=(8-1)/2

Y(8)=24+++

=990

Let find the function,

P==

y(x)=24+

=x3+6x2+11x+6

Y(8)=83+6(8)2+11(8)+6

***Newton Backward Difference formula of Interpolation:***

Yn(x)=y0+ pΔy0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | x | Y | yn | 2yn | 3yn | | 1 | 24 | 96 |  |  | | 3 | 120 | 120 | 48 | | 216 | | 5 | 336 | 168 |  | | 384 | | 7 | 720 |  |  | |  | | 8 | ? |  |  | |

h=2

so, p= ==0.5

Y(8)=24+++

=990

**Source Code:**

#include <iostream>

#include<bits/stdc++.h>

using namespace std;

int factp(int k)

{

int i,p=1;

for(i=1;i<=k;i++)

{

p=p\*i;

}

return p;

}

double forwrd(int m,double p)

{

int i;

double pr=1.0,q;

for(i=1;i<=m-1;i++)

{

pr=pr\*(p-i);

}

q=p\*pr;

return q;

}

double backwrd(int m,double p)

{

int h;

double pr=1.0,q;

for(h=1;h<=m-1;h++)

{

pr=pr\*(p+h);

}

q=p\*pr;

return q;

}

int main()

{

double y[50][50],x[50];

int n,i,j;

double p,e,s,d,sum=0,r,sum1=0,r1;

cout<<"Enter n=";

cin>>n;

cout<<"\nx0=";

cin>>s;

cout<<"\nmissing Point=";

cin>>e;

cout<<"\nDiff=";

cin>>d;

p=(e-s)/d;

cout<<"p="<<p<<endl;

cout<<"X"<<" "<<"Y"<<endl;

for(i=0;i<n;i++)

{

cin>>x[i]>>y[0][i];

}

cout<<"Forward Difference Table"<<endl;

cout<<" X"<<" \t"<<"Y"<<"\t";

for(i=2;i<=n;i++)

{cout<<"del\_y"<<i-1<<"\t";}

cout<<"\n";

for(i=1;i<n;i++)

{

for(j=0;j<n-i;j++)

{

y[i][j]=y[i-1][j+1]-y[i-1][j];

}

}

for(j=0;j<n;j++)

{

cout<<x[j]<<"\t";

for(i=0;i<n-j;i++)

{

cout<<y[i][j]<<"\t";

}

printf("\n");

}

for(i=2;i<=n-1;i++)

{

sum=sum+(forwrd(i,p)/(factp(i)))\*y[i][0];

}

r=y[0][0]+p\*y[1][0]+sum;

cout<<"Y("<<e<<") is="<<r<<endl;

cout<<"Backward Difference"<<endl;

cout<<" X"<<" \t"<<"Y"<<"\t";

for(i=2;i<=n;i++)

{cout<<"del\_y"<<i-1<<"\t";}

cout<<"\n";

for(i=1;i<n;i++)

{

for(j=n-1;j>=i;j--)

{

y[i][j]=y[i-1][j]-y[i-1][j-1];

}

}

for(j=0;j<n;j++)

{

cout<<x[j]<<"\t";

for(i=0;i<=j;i++)

{

cout<<y[i][j]<<"\t";

}

printf("\n");

}

for(i=2;i<=n-1;i++)

{

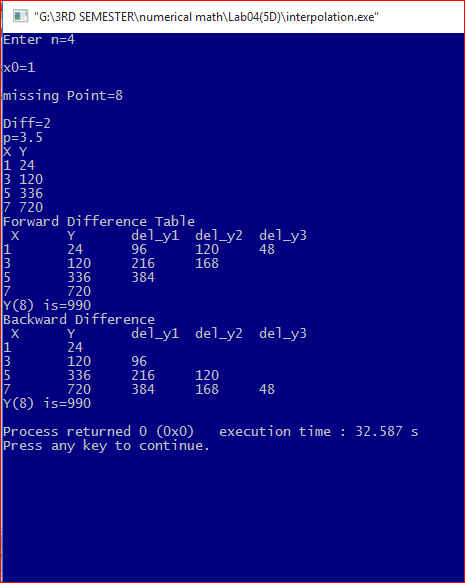
sum1=sum1+(backwrd(i,p)/(factp(i)))\*y[i][i];

}

r1=y[n-1][n-1]+p\*y[1][n-1]+sum1;

cout<<"Y("<<e<<") is="<<r<<endl;}

***Input/Output:***

******

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **Discussion:** This process is finding the value of y for some value of x outside the given range is called extrapolation and this example demonstrates the fact that if a tabulated function is a polynomial, then both interpolation and extrapolation would give exact value. Here y(8)=990 in both case(forward and backward both) |  |  |