Name : Deepankar Sharma

Course: BCA

University Roll No: 2092014

Student Id : 20041299

Semester: 3

Date: December 14, 2021

**Practical 12: Newton Backward Interpolation Formula**

**Objective:**To find value of y corresponding given value of x using Newton Backward Interpolation Formula.

**2. Algorithm:**

1. Start
2. Input the matrix of equations in arr[n][n+1] where n is number of given variables in data.

3. Calculate the difference table using:

For (j = 2; j < n+1; j++)

{

For (i = 0; i < n-j+1; i++)

{

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

}

}

4. Enter the value of x

5. Set value of p =x - mat[n-1][0];

p /= (mat[1][0] - mat[0][0])

6. p0=p

7. Find value of y using:

sum=0.0;

sum=sum+mat[n-1][1];

k=1;

for (j = 2; j < n+1; j++)

{

sum+= (p\*mat[n-1][j])/k;

k\*=j;

p\*=(p0+j-1);

}

8. y= sum

9. Print y

10. Stop

**Code:**

#include<iostream>

using namespace std;

int main(){

int n;

printf("Enter the number of known variables: ");

cin>>n;

// float mat[n][n+1]={0.0};

static float mat[10][10]={0.0};

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

{

/\* code \*/

printf("Enter x%d: ", i+1);

cin>>mat[i][0];

printf("Enter y%d: ", i+1);

cin>>mat[i][1];

}

cout<<"x"<<'\t'<<"y"<<endl<<endl;

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

{

/\* code \*/

cout<<mat[i][0]<<'\t'<<mat[i][1]<<endl;

}

// difference table

for (int j = 2; j < n+1; j++)

{

/\* code \*/

for (int i = n-1; i > j-2; i--)

{

/\* code \*/

// mat[i+1][j]= mat[i+1][j-1]- mat[i][j-1];

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

}

}

// displaying the difference table

cout<<"\nDisplaying the difference table: "<<endl<<endl;

cout<<"x"<<'\t'<<"y"<<'\t'<<"dy1"<<'\t'<<"dy2"<<'\t'<<"dy3"<<'\t'<<"dy4"<<'\t'<<"dy5";

cout<<endl<<endl;

int bVar = 2;

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

{

/\* code \*/

// if(i<2)

// for (int j = 0, j\_dash=n; j\_dash >= 0; j++, j\_dash--)

for (int j = 0; j<bVar; j++)

{

/\* code \*/

cout<<mat[i][j]<<'\t';

}

bVar++;

// else

cout<<endl;

}

// applying the Newton Backward formula

float x;

cout << "Enter the value of x at which you want to calculate the value of y: ";

scanf(" %f", &x);

float p = x - mat[n-1][0];

p /= (mat[1][0] - mat[0][0]);

float p0 = p;

float sum = 0.0;

sum = sum + mat[n-1][1];

int k = 1;

// for (int j = 2; j < n + 1; j++)

// {

// /\* code \*/

// sum += (p \* mat[0][j]) / k;

// k \*= j;

// p \*= (p0 - j + 1);

// }

for (int j = 2; j < n + 1; j++){

/\* code \*/

// sum += (p \* mat[0][j]) / k;

sum += (p \* mat[n-1][j]) / k;

k \*= j;

p \*= (p0 + j - 1);

}

printf("Value of y at x=%f is: %f\n\n", x, sum);

return 0;

}

**Output:**

PS E:\03 Semester\CBNST\Unit 03> cd "e:\03 Semester\CBNST\Unit 03\" ; if ($?) { g++ 12\_newtonBackwardInterpolation.cpp -o 12\_newtonBackwardInterpolation } ; if ($?) { .\12\_newtonBackwardInterpolation }

Enter the number of known variables: 5

Enter x1: 10

Enter y1: 600

Enter x2: 20

Enter y2: 512

Enter x3: 30

Enter y3: 439

Enter x4: 40

Enter y4: 346

Enter x5: 50

Enter y5: 243

x y

10 600

20 512

30 439

40 346

50 243

Displaying the difference table:

x y dy1 dy2 dy3 dy4 dy5

10 600

20 512 -88

30 439 -73 15

40 346 -93 -20 -35

50 243 -103 -10 10 45

Enter the value of x at which you want to calculate the value of y: 35

Value of y at x=35.000000 is: 395.429688