

Федеральное агентство связи
Федеральное государственное бюджетное образовательное учреждение
высшего образования «Сибирский государственный университет
телекоммуникаций и информатики»

Факультет: Информатики и вычислительной техники
Кафедра прикладной математики и кибернетики
Дисциплина: Вычислительная математика

Отчёт по лабораторной работе № 6
«Интерполяция кубическими сплайнами.»

Выполнила студентка группы ИА-831:
Угольников Екатерина Алексеевна
Проверил ассистент кафедры ПМиК:
Петухова Яна Владимировна

Новосибирск
2020

Задание

Написать программу, реализующую интерполяцию функции по набору точек, с помощью кубических сплайнов. Приложить решение конкретного примера этим методом.

Пример решения

Провести интерполяцию функции кубическими сплайнами.

Вычислить интерполяционный сплайн в точке $x=2$.

$$f(x) = \sqrt[3]{x}$$

$$f(2) = \sqrt[3]{2} = 1,25992$$

$$n=4$$

i	x	$y=f(x)$	$h=x_i-x_{i-1}$
0	1	1	0
1	1,3	1,09139	0,3
2	1,6	1,16961	0,3
3	1,8	1,21644	0,2
4	2,1	1,28058	0,3

Формула сплайна:

$$s_i(x) = M_{i-1} \frac{(x_i - x)^3}{6h_i} + M_i \frac{(x - x_{i-1})^3}{6h_i} + \left(y_{i-1} - \frac{M_{i-1}h_i^2}{6} \right) \frac{x_i - x}{h_i} + \left(y_i - \frac{M_i h_i^2}{6} \right) \frac{x - x_{i-1}}{h_i}$$

$$CM = D$$

$$C_{ij} = \begin{cases} \frac{h_i + h_{i+1}}{3}, & i=j \\ \frac{h_{i+1}}{6}, & j=i+1 \\ \frac{h_i}{6}, & j=i-1 \end{cases}$$

$$D_i = \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}$$

В нашем случае:

$$M = \begin{pmatrix} M_1 \\ M_2 \\ M_3 \end{pmatrix}$$

$$C = \begin{pmatrix} \frac{h_1+h_2}{3} & \frac{h_2}{6} & 0 \\ \frac{h_2}{6} & \frac{h_2+h_3}{3} & \frac{h_3}{6} \\ 0 & \frac{h_3}{6} & \frac{h_3+h_4}{3} \end{pmatrix} = \begin{pmatrix} \frac{0,3+0,3}{3} & \frac{0,3}{6} & 0 \\ \frac{0,3}{6} & \frac{0,3+0,2}{3} & \frac{0,2}{6} \\ 0 & \frac{0,2}{6} & \frac{0,2+0,3}{3} \end{pmatrix} = \begin{pmatrix} 0,2 & 0,05 & 0 \\ 0,05 & 0,166667 & 0,0333333 \\ 0 & 0,0333333 & 0,166667 \end{pmatrix}$$

$$D = \begin{pmatrix} \frac{y_2 - y_1}{h_2} - \frac{y_1 - y_0}{h_1} \\ \frac{y_3 - y_2}{h_3} - \frac{y_2 - y_1}{h_2} \\ \frac{y_4 - y_3}{h_4} - \frac{y_3 - y_2}{h_3} \end{pmatrix} = \begin{pmatrix} \frac{1,16961 - 1,09139}{0,3} - \frac{1,09139 - 1}{0,3} \\ \frac{1,21644 - 1,16961}{0,2} - \frac{1,16961 - 1,09139}{0,3} \\ \frac{1,20058 - 1,21644}{0,3} - \frac{1,21644 - 1,16961}{0,2} \end{pmatrix} = \begin{pmatrix} -0,043929 \\ -0,0265476 \\ -0,0203707 \end{pmatrix}$$

$$\begin{pmatrix} 0,2 & 0,05 & 0 \\ 0,05 & 0,166667 & 0,0333333 \\ 0 & 0,0333333 & 0,166667 \end{pmatrix} \cdot \begin{pmatrix} M_1 \\ M_2 \\ M_3 \end{pmatrix} = \begin{pmatrix} -0,043929 \\ -0,0265476 \\ -0,0203707 \end{pmatrix}$$

$$\left(\begin{array}{ccc|c} 0,2 & 0,05 & 0 & -0,043929 \\ 0,05 & 0,166667 & 0,0333333 & -0,0265476 \\ 0 & 0,0333333 & 0,166667 & -0,0203707 \end{array} \right)$$

Решим матрицу методом Гаусса и определим вектор коэффициентов M :

$$M = \begin{pmatrix} M_1 \\ M_2 \\ M_3 \end{pmatrix} = \begin{pmatrix} -0,200168 \\ -0,0779064 \\ -0,106643 \end{pmatrix}$$

Посчитаем сплайн.

$$2 \in [1,8; 2,1] \Rightarrow x \in [x_3; x_4]$$

$$s_4(x) = M_3 \frac{(x_4 - x)^3}{6h_4} + M_4 \frac{(x - x_3)^3}{6h_4} + \left(y_3 - \frac{M_3 h_4^2}{6} \right) \frac{x_4 - x}{h_4} + \left(y_4 - \frac{M_4 h_4^2}{6} \right) \frac{x - x_3}{h_4}$$

$$s_4(2) = -0,106643 \cdot \frac{(2,1 - 2)^3}{6 \cdot 0,3} + 0 \cdot \frac{(2 - 1,8)^3}{6 \cdot 0,3} + \left(1,21644 - \frac{-0,106643 \cdot 0,3^2}{6} \right) \frac{2,1 - 2}{0,3} +$$

$$+ \left(1,28058 - \frac{0 \cdot 0,3^2}{6} \right) \frac{2 - 1,8}{0,3} = -0,0000592461 + 0,406013 + 0,85372 = 1,2596737539$$

$$s_4(2) = 1,2596737539 \approx 1,25967$$

Ответ: $s_4(2) = 1,25967$.

Результат выполнения программы

```
gieko@Gieko:~/Рабочий стол/prog/HigherMath$ g++ -Wall -o VM6 VM6.cpp
gieko@Gieko:~/Рабочий стол/prog/HigherMath$ ./VM6
F(x)=pow(x,0.333334)
x      F(x)
1      1
1.3    1.09139
1.6    1.16961
1.8    1.21644
2.1    1.28058
X=2
F(X)=F(2)=1.25992
X=2 находится внутри отрезка [1.8;2.1]

Cubic Spline Interpolation:
S[4][X]=S[4][2]=1.25967
gieko@Gieko:~/Рабочий стол/prog/HigherMath$
```

Листинг программы

```
#include <iostream>
#include <algorithm>
#include <math.h>
#define ERROR -1
#define EPS 0.0000001
using namespace std;
int pos;
double f(double x){
    return pow(x,0.333334);
}
double** Transform(double **C,int n,int k){
    double max;
    short int in=k;
    max=abs(C[k][k]);
    for(short int i=k+1;i<n;i++){
        if(abs(C[i][k])>max){
            max=abs(C[i][k]);
            in=i;
        }
    }
    if(max<EPS){
        cout << "null column"<<endl;
        return NULL;
    }
    for(short int j=0;j<=n;j++) {
        double temp=C[k][j];
        C[k][j]=C[in][j];
        C[in][j]=temp;
    }
    return C;
}
double* Gauss(double **C, double *X, int n){
    int k=0;
    while(k<n){
        C=Transform(C,n,k);
        if(C==NULL) return NULL;
        else{
            for(int i=k;i<n;i++){
                double temp=C[i][k];
                if (abs(temp) < EPS) continue;
                for(short int j=0;j<=n;j++)
                    C[i][j]/=temp;
                if(i==k) continue;
                for(short int j=0;j<=n;j++)
                    C[i][j]-=C[k][j];
            }
            k++;
        }
        for(k=n - 1;k>=0;k--){
            X[k]=C[k][n];
            for(int i=0;i<k;i++)
                C[i][n]-=C[i][k]*X[k];
        }
        return X;
    }
}
int FindX(double *X,double x,int n){
```

```

        for(short int i=0;i<n;i++)
            if((x>=X[i])&&(x<=X[i+1])) return i+1;
        return ERROR;
    }
    double Cubic_Spline_Interpolation(double *X, double *A,
    double x,int n){
        double S;
        double *H= new double [n];
        for(short int i=1;i<n;i++) H[i]=X[i]-X[i-1];
        double *M = new double [n];
        double **C = new double *[n-2];
        for (short int i=0;i<n-2;i++) {
            C[i] = new double[n-1];
        }
        for (short int j=1; j<n-1; j++){
            for (short int k=1; k<n-1; k++){
                if(j==k) C[j-1][k-1]=(H[j]+H[j+1])/3;
                else if(k==j+1) C[j-1][k-1]=(H[j+1])/6;
                else if(k==j-1) C[j-1][k-1]=H[j]/6;
                else C[j-1][k-1]=0;
            }
        }
        for (short int i=0; i<n-2; i++){
            C[i][n-2]=(A[i+2]-A[i+1])/H[i+2]-(A[i+1]-A[i])/H[i+1];
        }
        M=Gauss(C,M,n-2);
        for (short int k=n-2; k>0; k--) M[k]=M[k-1];
        M[0]=0;M[n-1]=0;
        pos=FindX(X,x,n);
        S=0;
        S+=M[pos-1]*pow(X[pos]-x,3)/(6*H[pos]);
        S+=M[pos]*pow(x-X[pos-1],3)/(6*H[pos]);

        S+=(A[pos-1]-M[pos-1]*pow(H[pos],2)/6)*(X[pos]-x)/H[pos];
        S+=(A[pos]-M[pos]*pow(H[pos],2)/6)*(x-X[pos-1])/H[pos];
        return S;
    }

    int main(){
        int n=5; double x=2;
        double *F_x= new double [n];
        double X[n]={1,1.3,1.6,1.8,2.1};
        cout<<"F(x)=pow(x,0.333334)"<<endl;
        cout<<"x\tF(x)"<<endl;
        for(short int i=0;i<n;i++){
            F_x[i]=f(X[i]);
            cout<<X[i]<<"\t"<<F_x[i]<<endl;
        }
        cout<<"X="<<x<<endl<<"F(X)=F("<<x<<")="<<f(x)<<endl;
        double S=Cubic_Spline_Interpolation(X,F_x,x,n);
        cout<<"X="<<x<<" находится внутри отрезка ["<<X[pos-
1]<<";"<<X[pos]<<"]"<<endl<<endl;
        cout<<"Cubic Spline Interpolation:"<<endl;
        cout<<"S["<<pos<<"][X]=S["<<pos<<"]("<<x<<")="<<S<<endl;
        return 0;
    }

```