БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

ЛАБОРАТОРНАЯ РАБОТА № 5

Вычислительные методы алгебры

Сплайн-интерполирование  
 методом моментов

Манец Мария

2 курс 1 группа

Преподаватель :

Будник А. М.

**Постановка задачи**

Для заданной функции f(x) на равномерной сетке узлов построить интерполяционный кубический сплайн и вычислить его значение в контрольных точках x\*, x\*\*, x\*\*\*, а также оценить результат.

**Входные данные:**

Функция : , а=0.1

x\* =1.03(3), x\*\*=1.53(3), x\*\*\*=1.96(6)

|  |  |  |
| --- | --- | --- |
| *i* |  |  |
| 0 | 1,0 | 1.0291520691730114 |
| 1 | 1,1 | 1.1025032264499353 |
| 2 | 1,2 | 1.1708468696441585 |
| 3 | 1,3 | 1.2341320336373982 |
| 4 | 1,4 | 1.2924247536740816 |
| 5 | 1,5 | 1.3459143949774555 |
| 6 | 1,6 | 1.394919485176866 |
| 7 | 1,7 | 1.4398930685799418 |
| 8 | 1,8 | 1.4814276142316705 |
| 9 | 1,9 | 1.5202595231466 |
| 10 | 2,0 | 1.5572732940361786 |

**Алгоритм решения и формулы**

В общем виде алгоритм решения поставленной задачи выглядит так:

1. Строим S3(x)

2. Находим значения в заданных точках

Теоретический материал:

S3(x)=

Полученная система решается методом прогонки, а C1 и C2 рассчитываются по таким формулам:

С1 =

C2 = , i= 1,..,n-1

M0=Mn=0

**Листинг**

#include <iostream>

#include <fstream>

#include <math.h>

#include <iomanip>

#include <cmath>

using namespace std;

ofstream out("output.txt");

class Spline{

public:

const double coef = 1.5;

const int N =10;

double A, B, h;

double \*х,\*f,\*l,\*v,\*d,\*c,\*mom;

Spline() {

A = 1, B = 2, h = 0.1;

х = new double[N + 1]; //коэффициенты в системе

f = new double[N + 1];

l = new double[N];

v = new double[N];

d = new double[N + 1];

c = new double[N + 1];

mom = new double[N+1]; //моменты

}

void initSweepCoefs() {

for (int i = 0; i < N + 1; i++) {

х[i] = A + i \* h;

f[i] = coef \* exp(Spline::х[i]) + (1 - coef) \* sin(Spline::х[i]);

out << fixed << setprecision(1) << Spline::х[i];

out << fixed << setprecision(15) << " "<< f[i] << endl;

}

l[0] = 0;

v[N - 1] = 0;

c[0] = c[N] = 2;

d[0] = 2 \* (coef \* exp(A) - (1-coef) \* sin(A));

d[N] = 2 \* (2.1 \* exp(B) - (1-coef) \* sin(B));

for (int i = 0; i<N - 1; i++) {

l[i + 1] = v[i] = -0.5;

c[i + 1] = 2;

d[i + 1] = (3 / h)\*(((f[i + 2] - f[i + 1]) / h) - ((f[i + 1] - f[i]) / h));

}

}

//Построение i-го сплайна

double S(int i, double x, double\* mom) {

return (mom[i - 1] \* pow(Spline::х[i] - x, 3) / 6 \* h) + (mom[i] \* pow(x - Spline::х[i - 1], 3) / 6 \* h) +

((Spline::х[i] - x) / h)\*(f[i - 1] - ((mom[i - 1] \* pow(h, 2)) / 6)) + ((x - Spline::х[i - 1]) / h)\*(f[i] - ((mom[i] \* pow(h, 2)) / 6));

}

double \*solveMatrix()

{

double \*x = new double[N+1];

double m;

for (int i = 1; i < N+1; i++)

{

m = v[i] / c[i - 1];

c[i] = c[i] - m\*l[i - 1];

d[i] = d[i] - m\*d[i - 1];

}

x[N] = d[N] / c[N];

for (int i = N - 1; i >= 0; i--)

x[i] = (d[i] - l[i] \* x[i + 1]) / c[i];

return x;

}

};

int main() {

Spline \*spline = new Spline();

spline->initSweepCoefs(); //Инициализация значений

double \*moments = spline->solveMatrix(); //Левая прогонка - нахождение моментов

double \*xch = new double[3];

double \*fch = new double[3];

double \*Rtrue = new double[3];

double \*Sch = new double[3];

xch[0] = spline->х[0] + (spline->h / 3);

xch[1] = spline->х[5] + (spline->h / 3);

xch[2] = spline->х[10] - (spline->h / 3);

out << endl << "x\* : "<< xch[0]

<< endl << "x\*\* : " << xch[1]

<< endl << "x\*\*\*: " << xch[2] << endl;

int ind[3] = { 1, 6, 10 };

out << endl;

//Значения функции в точках, построение сплайна, подсчет погрешности

for (int i = 0; i < 3; i++) {

fch[i] = spline->coef \* exp(xch[i]) + (1-spline->coef) \* sin(xch[i]); //fch - функция в точках

Sch[i] = spline->S(ind[i], xch[i], moments);

//sch - сплайн в точках

Rtrue[i] = abs(fch[i] - Sch[i]);

//Rtrue - погрешность = fch-sch

out << "f[" << i << "]: " << fch[i] << " "

<< "S[" << i << "]: " << Sch[i] << " "

<< "Pogr[" << i << "]: " << Rtrue[i] << endl;

}

return 0;

}

**Результаты и вывод**

|  |  |  |  |
| --- | --- | --- | --- |
|  | X |  |  |
| x\* | 1.033333333333333 | 1.0545097494756177 | 3.586620623896586E-4 |
| x\*\* | 1.533333333333333 | 1.3636529754683036 | 9.248227409623944E-4 |
| x\*\*\* | 1.966666666666667 | 1.5452250953247328 | 1.4852187645808534E-4 |

**Вывод:**

Точность результатов, полученных сплайн-интерполированием ниже, чем при помощи многочленов Лагранжа и Ньютона, так как степень многочлена на каждом отрезке ниже, а расстояние между соседними узлами недостаточно мало, чтобы получить более точный результат.