|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ДИСЦИПЛИНА «Вычислительные алгоритмы»

**Лабораторная работа № \_**3**\_\_**

|  |  |
| --- | --- |
| **Тема \_\_\_**Интерполяция функций кубическим сплайном**\_\_\_\_\_\_\_\_\_\_\_\_**  **Студент \_**Блохин Д.М.\_\_\_\_\_\_  **Группа \_**ИУ7-42Б**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель \_\_**Градов В.М.\_\_\_\_\_\_\_ |  |

Москва.

2020 г.

**1. Задание**

Построить алгоритм и программу кубической сплайн-интерполяции.

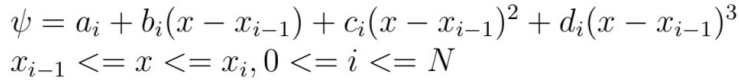
**2. Входные данные/Выходные данные**

Входные: Таблица функции с количеством узлов N, значение аргумента x.

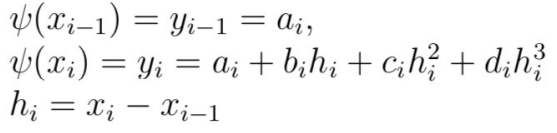
Выходные: Значения y(x).

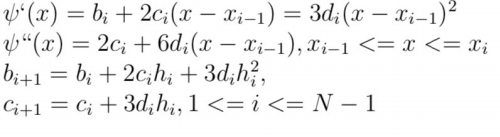
**3. Описание алгоритма**

**3.1 Интерполяционный полином на участке между каждой парой соседних точек имеет третью степень:**

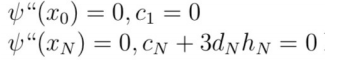


В узлах значения многочлена и интерполируемой функции совпадают:



Всего таких уравнений в два раза меньше чем число неихвестных. Чтобы найти недостающие уравнения, первые и вторые производные, вычисляемые по коэффициентам на соседних участках, приравниваются. 

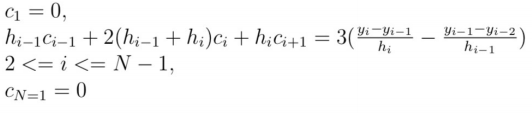
Недостающие условия можно получить, полагая, что вторая производная равна нулю на концах участка интерполяции.



Вышеописанные уравнения позволяют определить все 4N неизвестных коэффициентов:



В итоге получим систему уравнений:



**4.Метод прогонки**

Суть метода в следующем: применение метода Гаусса для решения системы уравнений с трехдиагональной матрицей приводит к тому, что система уравнений преобразуется к виду, когда в каждом уравнении содержится только два неизвестных и при обратном ходе одно из этих неизвестных выражается через другое. Поэтому можно записать:



Где ξi+1,ηi+1 - некоторые неизвестные пока прогоночные коэффициенты: Подставляя последнее выражение, получим:



Сравнивая два последних уравнения получим:



В этих формулах введено обозначение:



Из условия c1 = 0 следует ξ2 =0,η2 =0

**Код программы**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#define PI 3.14159265359

#define SUCCESS 0

double f(double x)

{

return sin(x);

}

double fs(double x)

{

return cos(x);

}

double fss(double x)

{

return -sin(x);

}

void file\_f\_func(char \*file\_name)

{

FILE \*file = fopen(file\_name, "wt");

for (double x = 0; x <= 4; x += 1)

{

fprintf(file, "%lf %lf\n", x, f(x));

}

fclose(file);

return;

}

void file\_fs\_func(char \*file\_name)

{

FILE \*file = fopen(file\_name, "wt");

for (double x = 0; x <= 2 \* PI; x += PI)

{

fprintf(file, "%lf %lf\n", x, fs(x));

}

fclose(file);

return;

}

void file\_fss\_func(char \*file\_name)

{

FILE \*file = fopen(file\_name, "wt");

for (double x = 0; x <= 4; x++)

{

fprintf(file, "%lf %lf\n", x, fss(x));

}

fclose(file);

return;

}

void file\_points\_read(double \*x, double \*y, double \*h, int \*f\_cnt, char \*file\_name)

{

FILE \*file = fopen(file\_name, "rt");

while (1)

{

if (feof(file))

break;

fscanf(file, "%lf %lf", x + \*f\_cnt, y + \*f\_cnt);

if (\*f\_cnt >= 2)

h[\*f\_cnt - 1] = x[\*f\_cnt - 1] - x[\*f\_cnt - 2];

if (feof(file))

break;

(\*f\_cnt)++;

}

}

int main(void)

{

char file\_f[64] = "x\_y.txt";

file\_f\_func (file\_f);

double f\_x [64];

double f\_y [64];

double a [64] = {0, };

double b [64] = {0, };

double c [64] = {0, };

double d [64] = {0, };

double h[64] = {0, };

double ksi [256] = {0, };

double Etta[256] = {0, };

double f\_i;

int f\_x\_cnt = 0;

file\_points\_read(f\_x, f\_y, h, &f\_x\_cnt, file\_f);

int N = f\_x\_cnt - 1;

for(int i = 2; i <= N; i++)

{

f\_i = 3 \* ((f\_y[i] - f\_y[i - 1]) / h[i] -

(f\_y[i - 1] - f\_y[i - 2]) / h[i - 1]);

ksi[i + 1] = - h[i] /

(h[i - 1] \* ksi[i] + 2 \* (h[i - 1] + h[i]));

Etta[i + 1] = (f\_i - h[i - 1] \* Etta[i]) /

(h[i - 1] \* ksi[i] + 2 \* (h[i - 1] + h[i]));

printf("%lf\n", (f\_i));

}

for(int i = N + 1; i >= 2; i--)

{

c[i - 1] = ksi[i] \* c[i] + etta[i];

}

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

a[i] = f\_y[i - 1];

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

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

printf("%lf %lf\n", f\_y[i] - f\_y[i - 1], c[i + 1] - 2 \* c[i]);

}

double x = 2.7;

int i = 1;

while(!(x >= f\_x[i - 1] && x <= f\_x[i]) && i < f\_x\_cnt - 1)

i++;

int j = i;

printf("Result:%lf\n", a[j] + b[j] \* (x - f\_x[i - 1]) +

c[j] \* pow((x - f\_x[i - 1]), 2) +

d[j] \* pow((x - f\_x[i - 1]), 3));

return SUCCESS;

}

**Контрольные вопросы**

1. Выписать значения коэффициентов сплайна, построенного на двух точках. A1 = y0 b1 = (y1 – y0 ) / (x1 – x0 ) c1 = 0 d1 = 0

2. Выписать все условия для определения коэффициентов сплайна, построенного на 3-х точках.

2.1)Вторые производные на концах отрезка интервала равны нулю

2.2)$y` = y`` в x2 (непрерывность)

2.3)Для каждого полинома (их 2) должны быть 2 точки-условия, то есть P1 - x1 и x2, P2 - x2 и x3

3. Определить начальные значения прогоночных коэффициентов, если принять, что для коэффициентов сплайна справедливо C1=C2. c1 = c2 \* ξ2 + η2 = c2 => η2 = 0, ξ2 = 1

4. Написать формулу для определения последнего коэффициента сплайна СN, чтобы можно было выполнить обратный ход метода прогонки, если задано kCN-1+mCN=p, где k, m и p - заданные числа. CN = (p - kCN-1) / m