

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. В. Г. ШУХОВА» (БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и
автоматизированных систем

Лабораторная работа №2

по дисциплине: «Вычислительная математика»

тема: Интерполяция функций

Выполнил: ст. группы ПВ-211

Чувилко Илья Романович

Проверил:

Бондаренко Татьяна Владимировна

Белгород 2023 г.

Цель работы: изучить понятие интерполяции, основные свойства интерполяционной функции, способы задания интерполяционной функции; понятие интерполяционного многочлена; изучить способы построения интерполяционного многочлена для случая равномерной и неравномерной сетки интерполяции; получить практические навыки решения задачи интерполяции с помощью ЭВМ.

Задания к работе

1. Найти область допустимых значений переменной x для функции $y = f(x)$ задания соответствующего варианта.
2. Составить таблицу значений функции $y = f(x)$, используя ($n \geq 6$) узлов интерполяции ($x_i \neq a$, где a точка, не являющаяся узлом интерполяционной сетке, в которой необходимо приближенно вычислить значение функции в соответствии с вариантом задания; $x_0 < a < x_n$).
3. По полученной таблице значений функции $y = f(x)$ составить интерполяционный многочлен Лагранжа для случаев линейной, квадратичной и кубической интерполяции: $L_1(x)$, $L_2(x)$, $L_3(x)$.
Замечание. Интервал (x_0, x_n) , $n = 1, 2, 3$, используемый для построения интерполяционного многочлена Лагранжа должен содержать точку a .
4. По таблице значений функции составить интерполяционный многочлен Ньютона $I_n(x)$. При построении интерполяционного многочлена Ньютона необходимо использовать конечные разности для случая равномерной сетки интерполяции и разделенные разности для неравномерной сетки интерполяции. Можно построить таблицу значений функции для равномерной сетки, выполнить построение многочлена Ньютона с конечными разностями, затем убрать 1 значение из середины таблицы и выполнить построение многочлена Ньютона с разделенными разностями для получившейся неравномерной сетки интерполяции.
5. Вычислить точное значение функции $y = f(x)$ при $x=a$ ($y_T = f(a)$).
6. Вычислить приближенное значение функции при $x=a$ по всем полученным интерполяционным многочленам.
7. Определить абсолютную Δ и относительную δ погрешность вычисления значения функции для каждого интерполяционного многочлена (интерполяционные многочлены Лагранжа и Ньютона) при

заданном значении $x=a$.

8. Построить в одной системе координат графики полученных интерполяционных функций (многочлены Лагранжа и Ньютона), исходной функции $y=f(x)$ и отметить значения функций в точке $x=a$.

9. Представить полученные результаты в виде таблицы (см. Табл. 2.1).

Содержание отчета

1. Титульный лист.
2. Цель работы. Вариант задания.
3. Текст задания к работе.
4. Выполнение задания соответствующего варианта вручную полностью. Все действия выполняемые при решении задачи вручную расписывать подробно с указанием всех промежуточных операций. Указание только окончательного ответа не допускается.
5. Текст программы, включающий необходимые комментарии и спецификации подпрограмм.
6. Результаты работы программы.

Вариант 23

23	$x + \lg x$	15
----	-------------	----

$$y = x + \lg x$$

1) D D 3: $x \in (0; +\infty)$

2) i x_i $f(x_i)$

0 12 13.08

1 13 14.119

2 14 15.146

3 16 17.204

4 17 18.23

5 18 19.255

3) i x_i $f(x_i)$ Линейная интерполяция

0 14 15.146 ✓

1 16 17.204

$$L_1(x) = y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0}$$

$$L_1(15) = 15.146 \left(\frac{15 - 14}{14 - 16} \right) + 17.204 \left(\frac{15 - 16}{16 - 14} \right) =$$

$$= 15.146 \cdot \frac{1}{2} + 17.204 \cdot \frac{1}{2} = 16.175.$$

i	x_i	$f(x_i)$
0	13	14.114
1	14	15.146
2	16	17.204

Квадратичная
чимеромодель

$$L_2(x) = y_0 \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + y_1 \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} +$$

$$+ y_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

$$L_2(15) = 14.114 \cdot \frac{(15-14)(15-16)}{(13-14)(13-16)} + \frac{15.146(15-13)}{(14-13)(14-16)} +$$

$$+ \frac{17.204 \cdot (15-13)}{(16-13)(16-14)} = 14.114 - \frac{1 \cdot (-1)}{2(1-3)} +$$

$$+ 15.146 \cdot \frac{2 \cdot (-1)}{1 \cdot (-2)} + 17.204 \cdot \frac{2 \cdot 1}{3 \cdot 2} = \underline{\underline{-14.116}} \quad \underline{\underline{8.12}}$$

i	x_i	$f(x_i)$
0	13	14.114
1	14	15.146
2	16	17.206
3	17	18.230

Кубическая
чимеромодель

$$L_3(x) = y_0 \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} + y_1 \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} +$$

$$+ y_2 \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} + y_3 \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}$$

$$L_3(15) = 14,114 \cdot \frac{(15-14)(15-16)(15-17)}{(13-14)(13-16)(13-17)} +$$

$$+ 15,146 \cdot \frac{(15-13)(15-16)(15-17)}{(14-13)(14-16)(14-17)} + \frac{15-13}{17-206} \cdot \frac{(15-13)(15-14)(15-17)}{(16-13)(16-14)(16-17)} +$$

$$+ 18,230 \cdot \frac{(15-13)(15-14)(15-16)}{(17-13)(17-14)(17-16)} = 14,114 \cdot \frac{1 \cdot (-1) \cdot (-2)}{(-1)(-3)(-4)} +$$

$$+ 15,146 \cdot \frac{2 \cdot (-1) \cdot (-2)}{1 \cdot (-2) \cdot (-3)} + 17,206 \cdot \frac{2 \cdot 1 \cdot (-2)}{3 \cdot 2 \cdot (-1)} + \frac{18,23 \cdot 2 \cdot 1 \cdot (-1)}{4 \cdot 3 \cdot 1} =$$

$$= \frac{14,114}{6} + \frac{15,146 \cdot 2}{3} + \frac{17,206 \cdot 2}{6} + \frac{18,23 \cdot (-1)}{6} = \underline{\underline{20,887}}$$

4. Метод конечных разностей

i	x_i	y_i	$y(x_i, x_{i+1})$	$y(x_i, x_{i+1}, x_{i+2})$	$y(x_0, x_1, x_2, x_3)$
0	11	12,041	$13,08 - 12,041 =$ $\frac{1,039}{1,039}$	0,005	0,003
1	12	13,08	1,039	0,002	
2	13	14,114	1,032		
3	14	15,146			

$$I_3(t) = 12,041 + 13,08 \cdot t + 1,039 \cdot t^2 + 0,005 \cdot t(t-1) + 0,003 \cdot t(t-1)(t-2)$$

$$t(x) = \frac{x - x_0}{h} = \frac{x - 11}{1} = x - 11$$

$$t(15) = 15 - 11 = 4$$

$$\begin{aligned} I_3(15) &= 12.041 + 1,039 \cdot 4 + 0,005 \cdot 4 \cdot 3 + 0,003 \cdot 4 \cdot 3 \cdot 2 = \\ &= 16,329 \end{aligned}$$

$$5. y_T = f(15) = x + \lg x = 16,176$$

6. Знайдіть чиє вимірювання

$$7. \Delta h_1 = |h_1(15) - y_T| = |16,175 - 16,176| = 0,001$$

$$\delta h_1 = \frac{\Delta h_1 \cdot 100\%}{h_1(15)} = 0,006\%$$

$$\Delta h_3 = |I_3(15) - y_T| = |20,887 - 16,176| = 4,71$$

$$\delta h_3 = \frac{\Delta h_3 \cdot 100\%}{I_3(15)} = 0,12\%$$

$$\Delta I_3 = |I(15) - y_T| = |16,329 - 16,176| = 0,153$$

$$\delta I_{3x} = \frac{\Delta I_3 \cdot 100\%}{I(15)} = 0,936\%$$

$$\Delta L_2 = |L_2(15) - y_T| = |16,18 - 16,176| = 0,02$$

$$\delta L_2 = \frac{0,02 \cdot 100\%}{16,18} = 0,12\%$$

№8

	Многочлен Лагранжа			Многочлен Ньютона	
Порядок	Линейный интерполяции	квадратичный интерполяции	кубический интерполяции	разделение интервалов	коэф. разности
A	0,001	0,47	0,01	0,153	0,016
S	0,006	0,22	0,12	0,336	0,896

10. Составить программу, реализующую вычисление приближенного значения функции в произвольной точке путем построения интерполяционного многочлена Ньютона для случая равномерной и неравномерной сетки.

Код программы:

```
#include <iostream>
#include <fstream>
#include "vector"

#define EPS 0.00001

using namespace std;

typedef pair<float, float> Point;

//Сравнение двух вещественных чисел
bool fcmp(float f1, float f2) {
    return (abs(f1 - f2) < EPS);
}

//Проверка сетки на равномерность
bool checkIfUniform(vector<Point> points) {
    for (int i = 1; i <= points.size() - 2; i++) {
        if (!fcmp(points[i].first - points[i - 1].first,
                  points[i + 1].first - points[i].first))
            return false;
    }

    return true;
}

//Получить вектор конечных разностей
vector<vector<float>> getFiniteDifferences(vector<Point> points) {
    vector<vector<float>> finiteDifferences(points.size());
    for (int i = 0; i < points.size(); i++) {
        finiteDifferences[i] = vector<float>{points[i].second};
    }
}
```

```

for (int column = 1; column <= points.size() - 1; column++) {
    for (int row = 0; row < points.size() - column; row++) {
        finiteDifferences[row].push_back(finiteDifferences[row + 1][column - 1] -
                                         finiteDifferences[row][column - 1]);
    }
}

return finiteDifferences;
}

//Получить вектор разделенных разностей
vector<vector<float>> getDividedDifferences(vector<Point> points) {
    vector<vector<float>> dividedDifferences(points.size());
    for (int i = 0; i < points.size(); i++) {
        dividedDifferences[i] = vector<float>{points[i].second};
    }

    for (int column = 1; column <= points.size() - 1; column++) {
        for (int row = 0; row < points.size() - column; row++) {
            dividedDifferences[row].push_back(
                (dividedDifferences[row + 1][column - 1] -
                 dividedDifferences[row][column - 1]) /
                (points[row + 1].first - points[row].first));
        }
    }
}

return dividedDifferences;
}

//Получить многочлен N степени с вычитанием
float getNPowerPolynomialNegative(int power, float value) {
    float result = 1;
    for (int i = 0; i < power; i++) {
        result *= (value - i);
    }

    return result;
}

//Получить многочлен N степени с прибавлением
float getNPowerPolynomialPositive(int power, float value) {
    float result = 1;
    for (int i = 0; i < power; i++) {
        result *= (value + i);
    }

    return result;
}

//Получить многочлен для метода разделенных разностей
float getPolynomialForDividedDifferences(int power, float value,
                                          vector<Point> points) {
    float result = 1;
    for (int i = 0; i < power; i++) {
        result *= value - points[i].first;
    }

    return result;
}

```

```

//Найти значение факториала
int factorial(int n) {
    int result = 1;
    for (int i = 2; i <= n; i++) {
        result *= i;
    }

    return result;
}

int main(int argc, char *argv[]) {
    //Выбор способа ввода
    char source;
    cin >> source;

    float substitutedValue; // Подставляемое значение
    vector<pair<float, float>> points;

    //Считывание из терминала
    if (source == 't') {
        int numPoints;
        cin >> numPoints;

        for (int i = 0; i < numPoints; i++) {
            Point point;
            cin >> point.first >> point.second;
            points.push_back(point);
        }
        cin >> substitutedValue;

        //Считывание из файла
    } else if (source == 'f') {
        string filePath;
        cin >> filePath;

        ifstream inputFile(filePath);
        if (inputFile.is_open()) {
            int numPoints;
            inputFile >> numPoints;
            for (int i = 0; i < numPoints; i++) {
                Point point;
                inputFile >> point.first >> point.second;
                points.push_back(point);
            }
            cin >> substitutedValue;

            inputFile.close();
        } else {
            cerr << "Can't open file";
            exit(1);
        }
    } else {
        cerr << "Wrong input";
        exit(1);
    }

    if (checkIfUniform(points)) {
        vector<vector<float>> finiteDifferences = getFiniteDifferences(points);

```

```

float value = 0;
if (substitutedValue >= points[0].first) {
    //Интерполяция вперед
    float t = (substitutedValue - points[0].first) /
        (points[1].first - points[0].first);

    for (int i = 0; i < finiteDifferences[0].size(); i++) {
        value += finiteDifferences[0][i] * getNPowerPolynomialNegative(i, t) /
            factorial(i);
    }
} else {
    //Интерполяция назад
    float q = (substitutedValue - points[points.size() - 1].first) /
        (points[1].first - points[0].first);
    for (int row = finiteDifferences.size() - 1; row >= 0; row--) {
        int column = finiteDifferences[0].size() - 1 - row;
        value += finiteDifferences[row][column] *
            getNPowerPolynomialPositive(column, q);
    }
}

cout << "\nf(a) = " << value;

//Метод разделенных разностей
} else {
    vector<vector<float>> dividedDifferences = getDividedDifferences(points);

    float value = 0;
    for (int column = 0; column < dividedDifferences[0].size(); column++) {
        value += dividedDifferences[0][column] *
            getPolynomialForDividedDifferences(column, substitutedValue,
                points);
    }
    cout << "\nf(a) = " << value;
}
}
}

```

Равномерная сетка, интерполяция назад:

```

D:\BGTU\ViсMat\Lab2\Code1\cmake-build-debug\Code1.exe
t
3
16 17.204
17 18.23
18 19.255
15

f(a) = 16.174
Process finished with exit code 0

```

Равномерная сетка, интерполяция вперед:

```
D:\BGTU\VicMat\Lab2\Code1\cmake-build-debug\Code1.exe
t
J
12 13.08
13 14.114
14 15.146
11

f(a) = 12.038
Process finished with exit code 0
```

Неравномерная сетка:

```
D:\BGTU\VicMat\Lab2\Code1\cmake-build-debug\Code1.exe
t
J
12 13.08
13 14.114
16 17.204
15

f(a) = 16.158
Process finished with exit code 0
```

Вывод: в ходе лабораторной работы мы изучили понятие интерполяции, основные свойства интерполяционной функции, способы задания интерполяционной функции; понятие интерполяционного многочлена; изучили способы построения интерполяционного многочлена для случая равномерной и неравномерной сетки интерполяции; получили практические навыки решения задачи интерполяции с помощью ЭВМ.