НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 - Программная инженерия

Дисциплина – Вычислительная математика

Лабораторная работа №5

Вариант №11

Выполнил: Мухсинов С.П

Группа: P3217

Преподаватель: Малышева Т.А

Санкт-Петербург

2024

Цель работы

Решить задачу интерполяции, найти значения функции при заданных значениях аргумента, отличных от узловых точек.

Задание

__**Обязательное задание (до 80 баллов)**__

Вычислительная реализация задачи:

1. Выбрать из табл. 1 заданную по варианту таблицу $y = f(x)$ (таблица 1.1 –таблица 1.5).
2. Построить таблицу конечных разностей для заданной таблицы. Таблицу отразить в отчете.
3. Вычислить значения функции для аргумента $X1$ (см. табл.1), используя первую или вторую интерполяционную формулу Ньютона. Обратить внимание какой конкретно формулой необходимо воспользоваться.
4. Вычислить значения функции для аргумента $X2$ (см. табл. 1), используя первую или вторую интерполяционную формулу Гаусса. Обратить внимание какой конкретно формулой необходимо воспользоваться.

Программная реализация задачи:

1. Исходные данные задаются тремя способами:
   a) в виде набора данных (таблицы x,y), пользователь вводит значения с клавиатуры.
   b) в виде сформированных в файле данных (подготовить не менее трех тестовых вариантов).
   c) на основе выбранной функции, из тех, которые предлагает программа, например, $\sin x$. Пользователь выбирает уравнение, исследуемый интервал и количество точек на интервале (не менее двух функций).
2. Сформировать и вывести таблицу конечных разностей.
3. Вычислить приближенное значение функции для заданного значения аргумента, введенного с клавиатуры, указанными методами (см. табл. Сравнить полученные значения.
4. Построить графики заданной функции с отмеченными узлами интерполяции и интерполяционного многочлена Ньютона/Гаусса (разными цветами).
5. Программа должна быть протестирована на различных наборах данных, в том числе и некорректных.

Используемые формулы и методы

Интерполяционные формулы Ньютона для равноотстоящих узлов

Узлы интерполирования $x_0, x_1, \dots, x_n$ называются равноотстоящими, если: $x_i - x_{i-1} = h = const$, где h - шаг интерполирования, $x_i = x_0 + ih$.

Конечные разности применяются для функций, заданных на равномерной сетке.

Конечные разности нулевого порядка совпадают со значениями функции в узлах: $f(x_i) = y_i$.

Конечными разностями первого порядка называют величины:

$$\Delta y_i = y_{i+1} - y_i, \; i = 0, 1, \dots, n-1$$

Конечными разностями второго порядка называют величины:

$$\Delta^2 y_i = \Delta y_{i+1} - \Delta y_i$$

Конечными разностями k-го порядка называют величины:

$$\Delta^k y_i = \Delta^{k-1} y_{i+1} - \Delta^{k-1} y_i$$

Интерполяционные формулы Ньютона для равноотстоящих узлов

Конечные разности можно выразить непосредственно через значения функции:

$$\Delta^2 y_0 = \Delta y_1 - \Delta y_0 = (y_2 - y_1) - (y_1 - y_0) = y_2 - 2y_1 + y_0$$

$$\Delta^3 y_0 = \Delta^2 y_1 - \Delta^2 y_0 = \cdots = y_3 - 3y_2 + 3y_1 - y_0.$$

Аналогично для любого k можно записать:

$$\Delta^k y_0 = y_k - k y_{k-1} + \frac{k(k-1)}{2!} y_{k-2} + \cdots + (-1)^k y_0$$

$$\Delta^k y_i = y_{k+i} - k y_{k+i-1} + \frac{k(k-1)}{2!} y_{k+i-2} + \cdots + (-1)^k y_i$$

Используя конечные разности, можно определить y_k:

$$y_k = y_0 + k\Delta y_0 + \frac{k(k-1)}{2!}\Delta^2 y_0 + \cdots + \Delta^k y_0$$

Интерполяционные формулы Ньютона для равноотстоящих узлов

Перейдем к построению интерполяционного многочлена Ньютона.

Этот многочлен запишем в виде:

$$N_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)(x - x_1)\ldots(x - x_n) \qquad (*)$$

Условие интерполяции $N_n(x_i) = y_i$ используем для нахождения коэффициентов многочлена:

$$N_n(x_0) = a_0 = y_0$$

$$N_n(x_1) = a_0 + a_1(x_1 - x_0) = a_0 + a_1 h = y_1$$

$$N_n(x_2) = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) = a_0 + 2a_1 h + 2a_1 h^2 = y_2$$

…………………

Найдем отсюда коэффициенты a_0, a_1, a_2:

$$a_0 = y_0, \qquad a_1 = \frac{y_1 - a_0}{h} = \frac{y_1 - y_0}{h} = \frac{\Delta y_0}{h}, \qquad a_2 = \frac{y_2 - a_0 - 2a_1 h}{2h^2} = \frac{y_2 - y_0 - 2\Delta y_0}{2h^2} = \frac{\Delta^2 y_0}{2h^2}$$

Общая формула имеет вид:

$$a_k = \frac{\Delta^k y_0}{k!\,h^k} k = 0, 1, \ldots, n$$

Подставляя эти выражения в (*), получим:

$$\boldsymbol{N_n(x) = y_0 + \frac{\Delta y_0}{h}(x - x_0) + \frac{\Delta^2 y_0}{2!\,h^2}(x - x_0)(x - x_1) + \cdots + \frac{\Delta^n y_0}{n!\,h^n}(x - x_0)(x - x_1)\ldots(x - x_{n-1})}$$

Интерполяционные формулы Ньютона для равноотстоящих узлов

Введем обозначение: $t = (x - x_0)/h$. Тогда получим формулу Ньютона, которая называется **первой интерполяционной формулой Ньютона для интерполирования вперед:**

$$\boldsymbol{N_n(x) = y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_0 + \cdots + \frac{t(t-1)\ldots(t-n+1)}{n!}\Delta^n y_0}$$

Полученное выражение может аппроксимировать функцию на всем отрезке изменения аргумента $[x_0, x_n]$. Однако более целесообразно (с точки зрения повышения точности расчетов) использовать эту формулу для $x_0 \le x \le x_1$. При этом за x_0 может приниматься любой узел интерполяции x_k. Например, для $x_1 \le x \le x_2$, вместо x_0 надо взять значение x_1. Тогда интерполяционный многочлен Ньютона:

$$\boldsymbol{N_n(x) = y_i + t\Delta y_i + \frac{t(t-1)}{2!}\Delta^2 y_i + \cdots + \frac{t(t-1)\ldots(t-n+1)}{n!}\Delta^n y_i} \quad (*)$$

Интерполяционную формулу (*) обычно используют для вычислений значений функции в точках левой половины отрезка.

Интерполяционные формулы Ньютона для равноотстоящих узлов

Для правой половины отрезка разности вычисляют справа налево: $t = (x - x_n)/h.$ Тогда получим формулу Ньютона, которая называется **второй интерполяционной формулой Ньютона для интерполирования назад:**

$$\boldsymbol{N_n(x) = y_n + t\Delta y_{n-1} + \frac{t(t+1)}{2!}\Delta^2 y_{n-2} + \cdots + \frac{t(t+1)\ldots(t+n-1)}{n!}\Delta^n y_0}$$

Вычислительная реализация

	x	y	№ варианта	X_1	X_2
Таблица 1.1	0,25	1,2557	1	0,251	0,402
	0,30	2,1764	6	0,512	0,372
	0,35	3,1218	11	0,255	0,405
	0,40	4,0482	16	0,534	0,384
	0,45	5,9875	21	0,272	0,445
	0,50	6,9195	26	0,551	0,351
	0,55	7,8359	31	0,294	0,437

$$X_1 = 0{,}255$$

$$X_2 = 0{,}405$$

Построим таблицу конечных разностей

№	x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$	$\Delta^5 y_i$	$\Delta^6 y_i$
0	0,25	1,2557	0,9207	0,0247	-0,0437	1,0756	-4,1277	10,1917
1	0,30	2,1764	0,9454	-0,0190	1,0319	-3,0521	6,0640	
2	0,35	3,1218	0,9264	1,0129	-2,0202	3,0119		
3	0,40	4,0482	1,9393	-1,0073	0,9917			
4	0,45	5,9875	0,9320	-0,0156				
5	0,50	6,9195	0,9164					
6	0,55	7,8359						

$X_1 = 0{,}255$ находится в левой половине отрезка, поэтому воспользуемся формулой Ньютона для интерполирования вперед:

$$h = 0{,}05$$

$$t = \frac{x - x_0}{h} = \frac{0{,}255 - 0{,}25}{0{,}05} = 0{,}1$$

$$N_6(x) =$$
$$y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_0 + \frac{t(t-1)(t-2)}{3!}\Delta^3 y_0 + \frac{t(t-1)(t-2)(t-3)}{4!}\Delta^4 y_0 + \frac{t(t-1)(t-2)(t-3)(t-4)}{5!}\Delta^5 y_0 + \frac{t(t-1)(t-2)(t-3)(t-4)(t-5)}{6!}\Delta^6 y_0$$

$$N_6(0{,}255) = 1{,}2557 + 0{,}1 \cdot 0{,}9207 + \frac{0{,}1 \cdot (-0{,}9)}{2} \cdot 0{,}0247 + \frac{0{,}1(-0{,}9)(-1{,}9)}{6} \cdot (-0{,}0437)$$
$$+ \frac{0{,}1(-0{,}9)(-1{,}9)(-2{,}9)}{24} \cdot 1{,}0756 + \frac{0{,}1(-0{,}9)(-1{,}9)(-2{,}9)(-3{,}9)}{120} \cdot (-4{,}1277) +$$
$$+ \frac{0{,}1(-0{,}9)(-1{,}9)(-2{,}9)(-3{,}9)(-4{,}9)}{720} \cdot (10{,}1917) = 1{,}12252$$

Воспользуемся многочленом Гаусса для расчета $X_2 = 0{,}405 > a = 0{,}4$

№	x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$	$\Delta^5 y_i$	$\Delta^6 y_i$
-3	0,25	1,2557	0,9207	0,0247	-0,0437	1,0756	-4,1277	10,1917
-2	0,30	2,1764	0,9454	-0,0190	1,0319	-3,0521	6,0640	
-1	0,35	3,1218	0,9264	1,0129	-2,0202	3,0119		
0	0,40	4,0482	1,9393	-1,0073	0,9917			
1	0,45	5,9875	0,9320	-0,0156				
2	0,50	6,9195	0,9164					
3	0,55	7,8359						

Первая интерполяционная формула Гаусса $(x > a)$

$$P_n(x) = y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_{-1} + \frac{(t+1)t(t-1)}{3!}\Delta^3 y_{-1} + \frac{(t+1)t(t-1)(t-2)}{4!}\Delta^4 y_{-2} + \frac{(t+2)(t+1)t(t-1)(t-2)}{5!}\Delta^5 y_{-2} \dots + \frac{(t+n-1)\dots(t-n+1)}{(2n-1)!}\Delta^{2n-1} y_{-(n-1)} + \frac{(t+n-1)\dots(t-n)}{(2n)!}\Delta^{2n} y_{-n}$$

$$t = \frac{x - x_0}{h} = \frac{0{,}405 - 0{,}4}{0{,}05} = 0{,}1$$

$$P_6(0{,}405) = 4{,}0482 + 0{,}1 \cdot 1{,}9393 + \frac{0.1 \cdot (-0.9)}{2} \cdot 1{,}0129 + \frac{0.1 \cdot (-0.9) \cdot 1{,}1}{6} \cdot (-2{,}0202) + \frac{0.1 \cdot (-0.9) \cdot 1{,}1 \cdot (-1{,}9)}{24} \cdot (-3{,}0521) + \frac{0.1 \cdot (-0.9) \cdot 1{,}1 \cdot (-1{,}9) \cdot (2{,}1)}{24} \cdot (6{,}0640) + \frac{0.1 \cdot (-0.9) \cdot 1{,}1 \cdot (-1{,}9) \cdot (2{,}1) \cdot (-2{,}9)}{24} \cdot 10{,}1917 = 4{,}20971$$

Программная реализация

Многочлен Лагранжа

```
public class Lagrange extends Method{

    protected Lagrange(double point, double[] x, double[] y) {
        super(point, x, y);
    }

    @Override
    public Result solve() {
        double ln = 0;
```

```
        for(int i = 0; i < n; i++){
            ln += y[i] * (numerator(i)/denominator(i));
        }
        return new Result(point, ln, MethodType.LAGRANGE);
    }

    private double numerator(int i){
        double s = 1;
        for(int j = 0; j < n; j++){
            if(j != i){
                s *= (point - x[j]);
            }
        }
        return s;
    }

    private double denominator(int i){
        double s = 1;
        for(int j = 0; j < n; j++){
            if(j != i){
                s *= (x[i] - x[j]);
            }
        }
        return s;
    }
}
```

Многочлен Ньютона с разделенными разностями

```
public class NewtonDivided extends Method{
    protected NewtonDivided(double point, double[] x, double[] y) {
        super(point, x, y);
    }

    @Override
    public Result solve() {
        double N = 0;
        double p = 1;
        for(int i = 0; i < n; i++){
            N += f(i, 0)*p;
            p *= (point - x[i]);
        }
        return new Result(point, N, MethodType.NEWTON_DIVIDED);
    }

    private double f(int degree, int i){
        if(degree == 0){
            return y[i];
        }
        return (f(degree - 1, i + 1) - f(degree - 1, i))/(x[i + degree] -
x[i]);
    }
}
```

Многочлен Ньютона с конечными разностями

```
public class NewtonFinite extends Method{
    protected NewtonFinite(double point, double[] x, double[] y) {
        super(point, x, y);
    }
```

```java
    @Override
    public Result solve() {
        printTable();
        final double h = x[1] - x[0];
        if((x[0] <= point)&&(point <= (x[n-1] + x[0])/2)){
            System.out.println("Используется первая формула");
            double t = (point - x[0])/h;
            return solveLeft(t);
        }else{
            System.out.println("Используется вторая формула");
            double t = (point - x[n-1])/h;
            return solveRight(t);
        }
    }

    private double f(int degree, int i){
        if(degree == 0){
            return y[i];
        }
        return f(degree - 1, i + 1) - f(degree - 1,  i);
    }

    public void printTable(){
        for(int i = 0; i < n; i++){
            for(int j = 0; j < n - i;j++){
                System.out.print(f(j, i) + " ");
            }
            System.out.println();
        }
    }

    public Result solveLeft(double t){
        double N = y[0];
        double p = 1;
        for(int i = 0; i < (n-1); i++){
            p *= (t - i)/((i+1));
            N += f(i + 1, 0)*p;
        }
        return new Result(point, N, MethodType.NEWTON_FINITE);
    }

    public Result solveRight(double t){
        double N = y[n-1];
        double p = 1;
        for(int i = 0; i < n - 1; i++){
            p *= (t + i)/((i+1));
            N += f(i + 1, (n - 2 - i))*p;
        }
        return new Result(point, N, MethodType.NEWTON_FINITE);
    }
}
```

Результаты работы программы

Входные данные:

```
5
0.1 1.25
0.2 2.38
0.3 3.79
0.4 5.44
0.5 7.14
0.47
```

Вывод:

```
Многочлен Лагранжа:
Результат интерполяции:
Координата X: 0.47
Значение, вычисленное в X: 6.642079375

Таблица конечных разностей:
1.25 1.13 0.28000000000000025 -0.04000000000000036 -0.15000000000000124
2.38 1.4100000000000001 0.2400000000000002 -0.19000000000000128
3.79 1.6500000000000004 0.04999999999998934
5.44 1.6999999999999993
7.14

Используется вторая формула
Многочлен Ньютона с конечными разностями:
Результат интерполяции:
Координата X: 0.47
Значение, вычисленное в X: 6.642079374999999
```

Входные данные:

```
4
0.15 1.25
0.2 2.38
0.33 3.79
0.47 5.44
0.22
```

Вывод:

```
Многочлен Лагранжа:
Результат интерполяции:
Координата X: 0.22
Значение, вычисленное в X: 2.7074813034188034

Введенные данные не имеют конечные разности.
Многочлен Ньютона с разделенными разностями:
Результат интерполяции:
Координата X: 0.22
Значение, вычисленное в X: 2.707481303418803
```

Вывод

В ходе выполнения лабораторной работы были изучены различные методы интерполяции функций, а также была написана программа, вычисляющая приближенное значение функции в заданной точке, используя формулы интерполяции Лагранжа и Ньютона.