ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

ЧИСЛЕННОЕ ИНТЕРПОЛИРОВАНИЕ И ДИФФЕРЕНЦИРОВАНИЕ
(Вариант 8)

Выполнил студент 3 курса ПМиИ
Ковшов Максим

Постановка задачи:
Пусть на отрезке [a;b] заданы точки $x_0, x_1, \dots, x_n$ и значения функции $y = f(x)$ в этих точках: $y_0 = f(x_0), \dots, y_n = f(x_n)$.
Интерполяционный многочлен Лагранжа:
$L_n(x) = \sum_{i=0}^{n} \quad y_i \frac{(x-x_0)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)}$.
Оценка погрешности формулы Лагранжа:
$|R_n(x)| \le \frac{M_{n+1}}{(n+1)!}|(x-x_0)(x-x_1)\dots(x-x_n)|$,
где $M_{n+1} = max_{[x_0,x_n]}|f^{(n+1)}(x)|$
Если надо вычислить не общее выражение $L_n(x)$, а лишь его значение на конкретном x, то используется интерполяционная схема Эйткена:
$L_{i,i+1}(x) = \frac{1}{x_{i+1}-x_i}\begin{vmatrix} y_i & x_i-x \\ y_{i+1} & x_{i+1}-x \end{vmatrix}$,
$L_{i,i+1,i+2}(x) = \frac{1}{x_{i+2}-x_i}\begin{vmatrix} L_{i,i+1} & x_i-x \\ L_{i+1,i+2} & x_{i+2}-x \end{vmatrix}$,
$L_{i,i+1,i+2,i+3}(x) = \frac{1}{x_{i+3}-x_i}\begin{vmatrix} L_{i,i+1,i+2} & x_i-x \\ L_{i+1,i+2,i+3} & x_{i+3}-x \end{vmatrix}$ и тд

Первая часть.
1) Известная функция y=f(x) задана таблицей, в которой приведены значения в узлах с некоторой точностью. Составить по таблице интерполяционный многочлен Лагранжа. Вычислить значение функции в заданной точке x аналитически и с помощью многочлена Лагранжа. Найти левую, правую и центральную производную в этой точке и ее точное значение. Оценить погрешность полученных результатов.

2) Функция y = f(x) задана таблицей (одинаковой для всех вариантов):

x_k	1,00	1,08	1,20	1,27	1,31	1,38
y_k	1,17520	1,30254	1,50946	1,21730	1,22361	1,23470

Пользуясь интерполяционной схемой Эйткена найти f(x*), заданной в точке x* последовательно, используя все значения из таблицы.

Точность вычислений определяется числом значащих цифр в условии.

Вторая часть. Численное дифференцирование.
Во второй части задания вычислить таблицу на отрезке [a;b] на равномерной сетке (5 узлов), и в этих узлах и в точке m найти значение первой производной функции по формулам 1-го (левая и правая) и 2-го порядка точности и значение второй производной по формулам 2-го порядка точности, где это возможно.

Значение функции в точке m получить интерполированием по всем 5 точкам. Во всех точках найти точные значения производных. Оценить погрешность. Результаты свести в таблицу. Точность - 4 значащих цифры.

	Численно				Точно	
x_k, m	$f`(x)$ слева	$f`(x)$ справа	$f`(x)$ центр	$f``(x)$	$f`(x)$	$f``(x)$

Для точности m, кроме этого, указать значение функции и интерполированное значение.

Условия.

1 часть.

1) $y = sinx, x = 1{,}6.$

x_k	1,5	2,0	2,5	3,5
y_k	0,99745	0,9093	0,59847	0,14112

2) $x^* = 1{,}026$

2 часть.

$$f(x) = ln\frac{x}{2}, a = 4{,}5; b = 10; m = 5{,}03.$$

Ход выполнения

Задание 1

Дано:

1) $y = sin(x), x = 1,6$

x_k	1,5	2,0	2,5	3,5
y_k	0,99745	0,9093	0,59847	0,14112

2) x*=1,026

Результаты:

1)

```
Значение в X по Лагранжу 1.00762
Аналитическое значение 0.999574
Ошибка метода Лагранжа 0.00877469
Левая производная -0.0291495
Правая производная -0.0292495
Средняя производная -0.0291995
```

2)

```
Значение в X по Эйткену 0.966696
```

Задание 2

Дано:

$$f(x) = ln\frac{x}{2}; a = 4.5;\ b = 10;\ m = 5.03;$$

	Численно				Точно	
x_k, m	$f`(x)$ слева	$f`(x)$ справа	$f`(x)$ центр	$f``(x)$	$f`(x)$	$f``(x)$
4,5	-	0.22222	-	-	0.222222	-0.0493827
5,6	0.178573	0.17857	0.178571	-0.0318877	0.178571	-0.0318878
6,7	0.149255	0.149253	0.149254	-0.0222767	0.149254	-0.0222767
7,8	0.128206	0.128204	0.128205	-0.0164365	0.128205	-0.0164366
8,9	0.11236	0.112359	0.11236	-0.0126247	0.11236	-0.0126247
10	0.100001	0.0999995	0.1	-0.00999998	0.1	-0.01

```
x = 4.5
Левая производная 0.222225
Правая производная 0.22222
Средняя производная 0.222222
Вторая производная (численно) -0.0493827
Первая производная (аналитически) 0.222222
Вторая производная (аналитически) -0.0493827

x = 5.6
Левая производная 0.178573
Правая производная 0.17857
Средняя производная 0.178571
Вторая производная (численно) -0.0318877
Первая производная (аналитически) 0.178571
Вторая производная (аналитически) -0.0318878

x = 6.7
Левая производная 0.149255
Правая производная 0.149253
Средняя производная 0.149254
Вторая производная (численно) -0.0222767
Первая производная (аналитически) 0.149254
Вторая производная (аналитически) -0.0222767

x = 7.8
Левая производная 0.128206
Правая производная 0.128204
Средняя производная 0.128205
Вторая производная (численно) -0.0164365
Первая производная (аналитически) 0.128205
Вторая производная (аналитически) -0.0164366

x = 8.9
Левая производная 0.11236
Правая производная 0.112359
Средняя производная 0.11236
Вторая производная (численно) -0.0126247
Первая производная (аналитически) 0.11236
Вторая производная (аналитически) -0.0126247

x = 10
Левая производная 0.100001
Правая производная 0.0999995
Средняя производная 0.1
Вторая производная (численно) -0.00999998
Первая производная (аналитически) 0.1
Вторая производная (аналитически) -0.01

m = 5.03
Левая производная 0.198809
Правая производная 0.198805
Средняя производная 0.198807
Вторая производная (численно) -0.0395243
Первая производная (аналитически) 0.198807
Вторая производная (аналитически) -0.0395243
Интерполированное значение m 0.922224
Аналитическое значение m 0.922273
```

```
Интерполированное значение m: 0.922224
Аналитическое значение m: 0.922273
```

Выводы: Сопоставление численных результатов с аналитическими данными позволяет оценить точность применяемых методов. Высокая точность методов отражается в небольших различиях между полученными и аналитическими значениями. Применение интерполяции позволяет приблизительно определить значения функции в точках, которые не входят в исходную сетку. Анализ различий между интерполированными значениями и точными значениями функции дает представление о точности проведенной интерполяции.

Все исходные тексты программ приводятся в Приложении

ПРИЛОЖЕНИЕ

Код программы

```
#include <iostream>
#include <vector>

#define h 0.0001

//Первое задание
double function(double x) {
    return sin(x);
}
double derivitive_6(double x) {
    return -sin(x);
}

//Второе задание
double function_2(double x) {
    return log(x/2);
}
double derivitive_2_1(double x) {
    return 1/x;
}
double derivitive_2_2(double x) {
    return -1 / pow(x,2);
}

double lagrange(std::vector<double> x_k, std::vector<double> y_k, double x) {
    double L = 0;
    for (int i = 0; i < x_k.size(); i++) {
        double s = 1;
        for (int j = 0; j < x_k.size(); j++) {
            if (i != j) {
                s *= (x - x_k[j]) / (x_k[i] - x_k[j]);
            }
        }
        s *= y_k[i];
        L += s;
    }
    return L;
}
double lagrange_2(std::vector<double> x_k, double x) {
    double L = 0;
    for (int i = 0; i < x_k.size(); i++) {
        double s = 1;
        for (int j = 0; j < x_k.size(); j++) {
            if (i != j) {
                s *= (x - x_k[j]) / (x_k[i] - x_k[j]);
            }
        }
```

```
            s *= function_2(x_k[i]);
            L += s;
        }
        return L;
    }

double factorial(double x) {
    double result = 1;
    for (int i = 1; i < x; i++)
        result *= 1;
    return result;
}

double error_lagrange(std::vector<double> x_k, double x) {
    int n = 4;
    double result = 1;
    for (int i = 0; i < n; i++) {
        result *= x - x_k[i];
        result *= derivitive_6(2.5) / factorial(n+1);
    }
    return abs(result);
}

double left_derivative(double x, double (*func)(double)) {
    return (func(x) - func(x - h)) / h;
}

double mid_derivative(double x, double (*func)(double)) {
    return (func(x + h) - func(x - h)) / (2 * h);
}
double right_derivative(double x, double (*func)(double)) {
    return (func(x + h) - func(x)) / h;
}

//Численная вторая производная
double second_derivative(double x, double (*func)(double)) {
    return (func(x - h) - 2 * func(x) + func(x + h)) / pow(h, 2);
}

double eitken(std::vector<double> x_k, double x, std::vector<double> L, int k1, int k2) {
    if (k2 - k1 != 1)
        return (eitken(x_k, x, L, k1, k2 - 1) * (x_k[k2] - x) - (eitken(x_k, x, L, k1 + 1, k2)
            * (x_k[k1] - x))) / (x_k[k2] - x_k[k1]);
    else
        return L[k1];
```

```
}

int main()
{
    setlocale(LC_ALL, "Russian");
    double x = 1.6;
    std::vector<double> x_k = { 1.5, 2.0, 2.5, 3.5 };
    std::vector<double> y_k = { 0.99745, 0.9093, 0.59847, 0.14112 };
    std::cout << "Значение в X по Лагранжу " << lagrange(x_k, y_k, x) << std::endl;
    std::cout << "Аналитическое значение " << function(x) << std::endl;
    std::cout << "Ошибка метода Лагранжа " << error_lagrange(x_k, x) << std::endl;
    std::cout << "Левая производная " << left_derivative(x, function) << std::endl;
    std::cout << "Правая производная " << right_derivative(x, function) << std::endl;
    std::cout << "Средняя производная " << mid_derivative(x, function) << std::endl;

    x = 1.026;
    x_k = { 1.00, 1.08, 1.20, 1.27, 1.31, 1.38 };
    y_k = { 1.17520, 1.30254, 1.50946, 1.21730, 1.22361, 1.23470 };
    int n = 6;

    std::vector<double> L(n - 1);

    for (int i = 0; i < n - 1; i++)
        L[i] = (y_k[i] * (x_k[i + 1] - x) - (y_k[i + 1] * (x_k[i] - x))) / (x_k[i + 1] - x_k[i]);

    std::cout << "Значение в X по Эйткену " << eitken(x_k, x, L, 0, n-1) << std::endl;

    double a = 4.5;
    double b = 10;
    double m = 5.03;
    x_k = { 4.5, 5.6, 6.7, 7.8, 8.9, 10 };

    for (double i : x_k) {
        std::cout << "\nx = " << i << std::endl;
        std::cout << "Левая производная " << left_derivative(i, function_2) << std::endl;
        std::cout << "Правая производная " << right_derivative(i, function_2) << std::endl;
        std::cout << "Средняя производная " << mid_derivative(i, function_2) << std::endl;
        std::cout << "Вторая производная (численно) " << second_derivative(i, function_2) << std::endl;
        std::cout << "Первая производная (аналитически) " << derivitive_2_1(i) << std::endl;
        std::cout << "Вторая производная (аналитически) " << derivitive_2_2(i) << std::endl;
    }

    std::cout << "\nm = " << m << std::endl;
    std::cout << "Левая производная " << left_derivative(m, function_2) << std::endl;
    std::cout << "Правая производная " << right_derivative(m, function_2) << std::endl;
    std::cout << "Средняя производная " << mid_derivative(m, function_2) << std::endl;
    std::cout << "Вторая производная (численно) " << second_derivative(m, function_2) << std::endl;
    std::cout << "Первая производная (аналитически) " << derivitive_2_1(m) << std::endl;
    std::cout << "Вторая производная (аналитически) " << derivitive_2_2(m) << std::endl;
    std::cout << "Интерполированное значение m " << lagrange_2(x_k, m) << std::endl;
    std::cout << "Аналитическое значение m " << function_2(m) << std::endl;


}
```