

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Поволжский государственный университет телекоммуникаций и
информатики»

Факультет информатики и вычислительной техники

Кафедра При

ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №11
Дисциплина: Численные методы

Выполнил: студент
При-21 Морзюков М.А.
Проверил(а):
Осанов В.А.

Самара 2024

Цель работы: изучить численное дифференцирование: дифференцирование с помощью сплайнов.

Вариант 11

$$\int_1^2 e^{-\left(x+\frac{1}{x}\right)} dx$$

Дифференцирование с помощью сплайнов: это метод интерполяции, который позволяет построить гладкую кривую, проходящую через заданные точки (узлы). В отличие от линейной или квадратичной интерполяции, кубические сплайны обеспечивают более гладкое приближение, так как они используют кусочно-кубические полиномы.

```

public static double[] cubicSpline(double[] x, double[] y) {
    int n = x.length - 1;
    double[] h = new double[n];
    double[] a = new double[n + 1];
    double[] b = new double[n];
    double[] c = new double[n + 1];
    double[] d = new double[n];

    for (int i = 0; i < n; i++) {
        h[i] = x[i + 1] - x[i];
        a[i] = y[i];
    }
    a[n] = y[n];

    double[] alpha = new double[n];
    for (int i = 1; i < n; i++) {
        alpha[i] = (3.0 / h[i]) * (y[i + 1] - y[i]) - (3.0 / h[i - 1]) * (y[i] - y[i - 1]);
    }

    double[] l = new double[n + 1];
    double[] mu = new double[n];
    double[] z = new double[n + 1];

    l[0] = 1.0;
    for (int i = 1; i < n; i++) {
        l[i] = 2.0 * (x[i + 1] - x[i - 1]) - h[i - 1] * mu[i - 1];
        mu[i] = h[i] / l[i];
        z[i] = (alpha[i] - h[i - 1] * z[i - 1]) / l[i];
    }
    l[n] = 1.0;
    z[n] = 0.0;

    for (int j = n - 1; j >= 0; j--) {
        c[j] = z[j] - mu[j] * c[j + 1];
        b[j] = (y[j + 1] - y[j]) / h[j] - h[j] * (c[j + 1] + 2.0 * c[j]) / 3.0;
        d[j] = (c[j + 1] - c[j]) / (3.0 * h[j]);
    }

    double[] coefficients = new double[n * 4];
    for (int i = 0; i < n; i++) {
        coefficients[i * 4] = a[i];
        coefficients[i * 4 + 1] = b[i];
        coefficients[i * 4 + 2] = c[i];
        coefficients[i * 4 + 3] = d[i];
    }

    return coefficients;
}

public static double evaluateSplineDerivative(double[] x, double[] coefficients, double xVal) {
    int n = x.length - 1;
    int i = n - 1;
    while (i > 0 && x[i] > xVal) {
        i--;
    }

    int idx = i * 4;

    double dx = xVal - x[i];
    return coefficients[idx + 1] + 2 * coefficients[idx + 2] * dx + 3 * coefficients[idx + 3] * dx * dx;
}

```

Результат выполнения программы

x	f(x)	f'(x)
1,0	0,13534	-0,00777
1,1	0,13411	-0,02119
1,2	0,13090	-0,04056
1,3	0,12628	-0,05141
1,4	0,12072	-0,05917
1,5	0,11456	-0,06363
1,6	0,10807	-0,06586
1,7	0,10145	-0,06635
1,8	0,09484	-0,06555
1,9	0,08836	-0,06395
2,0	0,08208	-0,06134

Пример в MATLAB online:

