**Условие**

```
 13  1/10      РЕШИТЬ КРАЕВУЮ ЗАДАЧУ для оду  второго порядка
           методом конечных разностей и прогонки
           y'' - y' -   5* y =   X*X +   2* X -   3
           для  -1 <= x <=   0  с шагом 0.20000 и краевыми условиями
            -  5* y(-1)  -   3* y'(-1)   =   2
            -  2* y( 0)  -   4* y'( 0)   =   3
```

**Теория**

# 13. Решение краевой задачи для ОДУ

Рассмотрим численное решение дифференциального уравнения второго порядка на отрезке $x \in [a;b]$ :

$$K(x) \cdot y'' + L(x) \cdot y' + M(x) \cdot y = F(x). \qquad (13.1)$$

С заданными краевыми (граничными) условиями третьего рода на границах отрезка ( при $x=a$ и $x=b$ ):

$$R \cdot y'(a) + S \cdot y(a) = T,$$
$$V \cdot y'(b) + W \cdot y(b) = Z. \qquad (13.2)$$

Краевыми условиями первого рода называют условия на функцию ( например, $S \cdot y(a) = T$ ). Краевыми условиями второго рода называют условия на производную функции ( например, $R \cdot y'(a) = T$ ). Смешанными краевыми условиями называют условия разного рода на левой и правой границах отрезка $[a;b]$ .

Коэффициенты $K$, $L$, $M$ уравнения (13.1) могут быть постоянными. Это не сильно упростит решение, поэтому рассмотрим более общий случай, когда они зависят от $x$, как и искомая функция $y(x)$.

## 13.1. Метод конечных разностей

Разделим отрезок $[a;b]$ $n$ точками на $n$-$1$ отрезков одинаковой длины $h=(b-a)/(n-1)$ точками $x_i=a+i \cdot h$ ( $x_1=a$, $x_2=a+h, \ldots, x_n=b$ ). Значение искомой функции в точке $x=x_i$ обозначим $y_i$. Целью вычислений является нахождение таблицы $x$–$y$. В этой таблице неизвестными являются $n$ значений $y_i$, значит для их нахождения необходимо найти $n$ уравнений. Два уравнения дадут краевые условия ( 13.2 ) и *n-2* уравнений получим, записав дифференциальное уравнение ( 13.1 ) в *n-2* внутренних точках ( $x_2$, $x_3$, …, $x_{n-1}$ ) отрезка [$a$;$b$]. Воспользуемся соотношениями:

$$y'(a) = \frac{y_2 - y_1}{h} + O(h) \quad , \quad y'(b) = \frac{y_n - y_{n-1}}{h} + O(h) \qquad (13.3)$$

для записи краевых условий ( 13.2 ) в *конечно-разностном* виде

$$R \cdot \frac{y_2 - y_1}{h} + S \cdot y_1 = T \quad \text{и} \quad V \cdot \frac{y_n - y_{n-1}}{h} + W \cdot y_n = Z \quad \text{или окончательно}$$

$$-\left(\frac{R}{h} - S\right) y_1 + \frac{R}{h} y_2 = T \qquad \text{и} \qquad \left(\frac{V}{h}\right) y_{n-1} - \left(\frac{V}{h} + W\right) y_n = -Z\ . \qquad (13.4)$$

Это первое и последнее ( $n$-е ) уравнения будущей системы ( СЛАУ ) с трёхдиагональной матрицей.

Остальные *n-2* уравнения получим из дифференциального уравнения, записывая его для каждой внутренней точки $x_i$ , $i$=2,3,…,*n-1* , заменяя $y$, $y'$, $y''$ в точке $x_i$ по формулам

$$y(x_i) = y_i\ , \ y'(x_i) = \frac{y_{i+1} - y_{i-1}}{2h} + O(h^2)\ , \ \ y''(x_i) = \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} + O(h^2)\ . \quad (13.5)$$

Итак, для каждой внутренней точки $x_i$ , $i$=2,3,…,*n-1* получим конечно-разностное соотношение

$$\left(\frac{K(x_i)}{h^2} - \frac{L(x_i)}{2h}\right) y_{i-1} - \left(\frac{2K(x_i)}{h^2} - M(x_i)\right) y_i + \left(\frac{K(x_i)}{h^2} + \frac{L(x_i)}{2h}\right) y_{i+1} = F(x_i) \qquad (13.6)$$

Это соотношение является уравнением ( с номером $i$ ) искомой СЛАУ с трёхдиагональной матрицей.

Если соберём вместе в систему уравнения ( 13-4 ) и ( 13-6 ), получим

$$\left(\begin{array}{ccccccccc|c}
b_1 & c_1 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & d_1 \\
a_2 & b_2 & c_2 & 0 & 0 & \ldots & 0 & 0 & 0 & d_2 \\
0 & a_3 & b_3 & c_3 & 0 & \ldots & 0 & 0 & 0 & d_3 \\
0 & 0 & a_4 & b_4 & c_4 & \ldots & 0 & 0 & 0 & d_4 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
0 & 0 & 0 & 0 & 0 & \ldots & b_{n-2} & c_{n-2} & 0 & d_{n-2} \\
0 & 0 & 0 & 0 & 0 & \ldots & a_{n-1} & b_{n-1} & c_{n-1} & d_{n-1} \\
0 & 0 & 0 & 0 & 0 & \ldots & 0 & a_n & b_n & d_n
\end{array}\right) \qquad (13.7)$$

В этой расширенной матрице

$b_1 = -R/h + S$, $\quad c_1 = R/h$, $\quad d_1 = T$,

$a_2 = K(x_2)/h^2 - L(x_2)/2/h$, $\quad b_2 = -2K(x_2)/h^2 + M(x_2)$, $\qquad$ (13.8)

$c_2 = K(x_2)/h^2 + L(x_2)/2/h$, $\quad d_2 = F(x_2)$, $\quad a_3 = K(x_3)/h^2 - L(x_3)/2/h$,

$b_3 = -2K(x_3)/h^2 + M(x_3)$, $\quad c_3 = K(x_3)/h^2 + L(x_3)/2/h$, $\quad d_3 = F(x_3)$, … ,

$a_n = V/h$, $\quad b_n = -V/h - W$, $\quad d_n = -Z$.

Решим эту систему методом прогонки ( см. раздел 1.3 ) и получим в качестве ответа значения $y_i$ ( $i$=1,…,$n$ ) табличной функции $x$-$y$.

Имеющуюся таблицу можно аппроксимировать теми методами, которые были рассмотрены ранее.

## Решение

*Код программы*:

KonechnieRaznosti.java

```
package fourth;

import first.Progonka;

abstract public class KonechnieRaznosti {
    abstract double K(double x);
    abstract double L(double x);
    abstract double M(double x);
    abstract double F(double x);
    abstract double R();
    abstract double S();
    abstract double W();
    abstract double V();
    abstract double T();
    abstract double Z();
    abstract double a();
    abstract double b();
    abstract double h();
    void exec(){
        int iCount = (int)((b() - a())/h()) + 1;
        double[] X = new double[iCount];
        for(int i = 0;i<iCount;++i){
            X[i] = a() + (double)i * h();
        }
        double[][] matrix = new double[iCount][3];
        double[] d = new double[iCount];
        matrix[0][0] = 0;
        matrix[0][1] = (-R()/h()) + S();
        matrix[0][2] = R()/h();
        d[0] = T();
        for(int i = 1;i<iCount-1;++i){
            double x = X[i];
            matrix[i][0] = (K(x)/(h()*h())) - L(x)/2.0/h();
            matrix[i][1] = (-2.0*K(x)/(h()*h())) + M(x);
            matrix[i][2] = (K(x)/(h()*h())) + L(x)/2.0/h();
            d[i] = F(x);
        }
        matrix[matrix.length-1][0] = V()/h();
        matrix[matrix.length-1][1] = (-V()/h()) - W();
        matrix[matrix.length-1][2] = 0;
        d[d.length-1] = -Z();
        System.out.println("Матрица: ");
        for(int i = 0;i< matrix.length;++i){
            double[] values = matrix[i];
            System.out.printf("a: %.9f, b: %.9f, c: %.9f, d: %.9f\n",
                    values[0],values[1],values[2], d[i]);
        }
        d[matrix.length-1] = -Z();
        double[] res = Progonka.exec(matrix.length, matrix, d);

        System.out.println("Результаты:");
        for(int i = 0;i<res.length;++i){
            System.out.printf("x: %.9f, y: %.9f\n", X[i], res[i]);
        }
    }
}
```

Progonka.java

```
package first;

public class Progonka {
    public static int n;
    public static double[][] matrix;
    public static double[] d;
    public static double[] x;
    public static double[] p, q;
    public static double getA(int index){
        return matrix[index-1][0];
    }
    public static double getB(int index){
        return matrix[index-1][1];
    }
    public static double getC(int index){
        return matrix[index-1][2];
    }
    public static double getD(int index){
        return d[index-1];
    }
    public static double[] exec(int mN, double[][] mAMatrix, double[]
mBMatrix){
        n = mN;
        matrix = mAMatrix;
        d = mBMatrix;
        p = new double[n+1];
        q = new double[n+1];
        x = new double[n];
        p[0] = q[0] = 0;
        System.out.println("P[i]\t\tQ[i]");
        printRes(p[0],q[0]);
        for(int i = 1;i<n+1;++i){
            p[i] = (-getC(i))/(getB(i) + getA(i)*p[i-1]);
            q[i] = (getD(i) - (getA(i)*q[i-1]))/(getB(i) + getA(i)*p[i-1]);
            printRes(p[i],q[i]);
        }
        x[n-1] = q[n];
        for(int i = n-2;i>=0;--i){
            x[i] = q[i+1] + p[i+1]*x[i+1];
        }
        for(int i = 0;i<n;++i){
            System.out.println("x[" + i+ "] : " + x[i]);
        }
        return x;
    }
    static void printRes(double p, double q){
        System.out.printf("%2.9f %2.9f\n", p,q);
    }
}
```

Main.java

```
package fourth;

public class Main {
    public static void main(String[] args){
        KonechnieRaznosti task0 = new KonechnieRaznosti() {
            @Override
```

```
            double K(double x) { return 1; }
            @Override
            double L(double x) { return -1; }
            @Override
            double M(double x) { return -5; }
            @Override
            double F(double x) { return x*x + 2*x-3; }
            @Override
            double R() { return -3; }
            @Override
            double S() { return -5; }
            @Override
            double W() { return -2; }
            @Override
            double V() { return -4; }
            @Override
            double T() { return 2; }
            @Override
            double Z() { return 3; }
            @Override
            double a() { return -1; }
            @Override
            double b() { return 0; }
            @Override
            double h() { return 0.2; }
        };

        task0.exec();
    }
}
```

## Вывод консоли:

```
Матрица:
a: 0,000000000, b: 10,000000000, c: -15,000000000, d: 2,000000000
a: 27,500000000, b: -55,000000000, c: 22,500000000, d: -3,960000000
a: 27,500000000, b: -55,000000000, c: 22,500000000, d: -3,840000000
a: 27,500000000, b: -55,000000000, c: 22,500000000, d: -3,640000000
a: 27,500000000, b: -55,000000000, c: 22,500000000, d: -3,360000000
a: -20,000000000, b: 22,000000000, c: 0,000000000, d: -3,000000000

P[i]        Q[i]
0,000000000 0,000000000
1,500000000 0,200000000
1,636363636 0,688000000
2,250000000 2,276000000
-3,272727273 -9,633454545
0,155172414 -1,803862069
-0,000000000 -2,067956204

x[0] : -7.9808244555709775
x[1] : -5.4538829703806515
x[2] : -3.753372926343732
x[3] : -2.6797213005972154
x[4] : -2.1247518248175203
x[5] : -2.067956204379564

Результаты:
x: -1,000000000, y: -7,980824456
x: -0,800000000, y: -5,453882970
x: -0,600000000, y: -3,753372926
```

```
x: -0,400000000, y: -2,679721301
x: -0,200000000, y: -2,124751825
x: 0,000000000, y: -2,067956204

Process finished with exit code 0
```

**Ответ:**

| X | Y |
|---|---|
| -1,000000000 | -7,980824456 |
| -0,800000000 | -5,453882970 |
| -0,600000000 | -3,753372926 |
| -0,400000000 | -2,679721301 |
| -0,200000000 | -2,124751825 |
| 0,000000000 | -2,067956204 |