МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Лабораторная работа №2

РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ МЕТОДОМ ПРОГОНКИ

Преподаватель: Полевиков Виктор Кузьмич
доцент кафедры вычислительной математики
Студент: Болтач Антон Юрьевич
2 курс 9 группа

Минск 2017

Постановка задачи

1) Построить стандартную программу решения систем линейных алгебраических уравнений методом прогонки.
2) Рассмотрим СЛАУ вида:

$$\mathbf{A*x = f}$$

Задана невырожденная трёхдиагональная матрица **A** размером 21x21:

Матрица A:

```
9.53 -74.52 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

0.0 -93.7 44.73 -82.74 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

0.0 0.0 6.05 92.9 -92.8 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

0.0 0.0 0.0 4.33 -40.9 -8.38 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0 -56.5 61.62 -38.15 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0 0.0 -83.37 84.79 80.75 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0 0.0 0.0 -18.82 -1.08 23.81 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0 0.0 0.0 0.0 29.8 -39.36 -14.92 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -15.23 -93.31 -86.07 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0

0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 72.8 36.09 -88.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -70.23 -58.02 -84.15 0.0 0.0 0.0 0.0 0.0 0.0
0.0

0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 69.18 36.93 -45.08 0.0 0.0 0.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -63.04 12.42 -21.3 0.0 0.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -80.6 -7.71 -38.79 0.0 0.0 0.0 0.0

0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 2.13 35.2 -10.42 0.0 0.0 0.0

0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -94.63 -19.6 77.84 0.0 0.0

0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 11.78 -64.5 -44.62 0.0

0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -87.79 -79.33 71.25

0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 -87.19 74.69

-0.39 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 63.39
```

Транспонированный вектор значений f:

```
2305.398  -619.082 -3258.0123 -2127.262 2125.654 -6171.102 117.860 -150.191 -2000.416
-7095.394 -1012.336 -3867.979 -2212.326 32.412 -476.383 -2684.778 4666.162 -3241.677
-3739.265 -4290.555 -7343.812
```

Транспонированный вектор значений точного решения x:

```
29.07   1.97   24.65   -65.85   -37.95   -69.41   15.88   1.23   21.72   71.71   -97.84   95.02
6.79  -3.14  -3.69  -40.46  -74.55  -3.32  -81.95  -90.32  16.51
```

Краткая теория и алгоритм решения

Пусть дана системаа линейных уравнений с невырожденной трехдиагональной матрицей:

$$\begin{pmatrix} C_0 & -B_0 & 0 & 0 & 0 & 0 & \dots & 0 \\ -A_1 & C_1 & -B_1 & 0 & 0 & 0 & \dots & 0 \\ 0 & -A_2 & C_2 & -B_2 & 0 & 0 & \dots & 0 \\ & & & \dots & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & A_{N-1} & C_{N-1} \end{pmatrix}$$

Такую систему можно записать в виде:

$$C_0 y_0 - B_0 y_1 = f_{0,}, i = 0$$

$$-A_i y_{i-1} + C_i y_i - B_i y_{i+1} = f_i, \ 1 \le i \le N-1$$

$$-A_N y_{N-1} + C_N y_N = f_N\,, i = N$$

Алгоритм можно условно разделить на два этапа:

- Прямой ход
- Обратный ход

<u>Прямой ход метода прогонки</u>

Находим α_1 и β_1: $\alpha_1 = \frac{B_0}{C_0}$, $\beta_1 = \frac{f_0}{C_0}$.

Вычисляем α_i, β_i для $i = 1, 2, 3, \dots, N$-1:

$$\alpha_{i+1} = \frac{B_i}{C_i - \alpha_i A_i}; \beta_{i+1} = \frac{f_i + \beta_i A_i}{C_i - \alpha_i A_i}.$$

<u>Обратный ход метода прогонки</u>

Находим $y_N = \frac{f_N + \beta_N A_N}{C_N - \alpha_N A_N}$.

Вычисляем $y_{N-1}, y_{N-2}, \dots, y_0$ по формулам:

$$y_i = \alpha_{i+1} y_{i+1} + \beta_{i+1}, \qquad i = \overline{N-1, 0}$$

<u>Применимость и устойчивость метода прогонки</u>

Метод прогонки применим и устойчив к округлениям, если выполнены следующие условия:

1) $C_0 \neq 0$, $C_N \neq 0$; $A_i \neq 0$, $B_i \neq 0$ при $i = \overline{1, N-1}$.
2) $|C_i| \ge |A_i| + |B_i|$ при $i = \overline{1, N-1}$.
3) $|C_0| \ge |B_0|$, $|C_N| \ge |A_N|$ (причем хотя бы одно из неравенств должно выполняться строго, то метод применим и устойчив)

Листинг программы на языке программирования Java

```
import java.util.Random;


public class method {
    public static void main(String[] args) {
        int n = 20;
        double arr[][] = new double[n+1][n+1];
        double a[] = new double[n+1]; // sub
        double c[] = new double[n+1]; // main
        double b[] = new double[n+1]; // sup
        Random r = new Random();
        for (int i = 1; i <= n; i++) {
          for (int j = 1; j <= n; j++) {
            if(i == j || j == i + 1 || j == i - 1) {
              arr[i][j] = -100 + (200) * r.nextDouble();
              arr[i][j] = Math.rint(arr[i][j] * 100.0) / 100.0;
              c[j] = arr[i][j];
              b[j] = arr[i][j];
              a[j] = arr[i][j];
            } else
              arr[i][j] = 0;
            System.out.print(arr[i][j] + " ");
            if (j == n - 1)
              System.out.println();
          }
        }
        double x[] = new double[n+1];
        for (int i = 1; i <= n; i++) {
          x[i] = -100 + (200) * r.nextDouble();
          x[i] = Math.rint(x[i] * 100.0) / 100.0;
        }
        double f[] = new double[n+1];
        for (int j = 1; j <= n; j++)
            for (int i = 1; i <= n; i++)
                f[j] += arr[i][j] * x[i];
        System.out.println();
        for(int i = 1; i <= n; i++)
                System.out.println(f[i]);
```

```
        System.out.println();
        double answer[] = new double[n+1];
        answer = Prorace(arr, n, f);
        for (int i = 1; i <= n; i++)
          System.out.println(x[i] + "\t" + answer[i]);
        System.out.println();
    }
    static double[] Prorace(double[][] a, int n, double[] b)
    {
        double k[] = new double[n+1]; // коэффициенты при Ci
        double m[] = new double[n+1]; // коэффициенты при Ai
        double t[] = new double[n+1]; // коэффициенты при Bi
        double p[] = new double[n+1]; // коэффициенты α
        double q[] = new double[n+1]; // коэффициенты β
        double y[] = new double[n+1]; // вектор решения y
        for (int i = 1; i <= n; i++) { // заполнение векторов
            if (i == 1) k[i] = 0;
            else k[i] = a[i][i - 1];
            m[i] = -a[i][i];
            if (i == n ) t[i] = 0;
            else
                t[i] = -a[i][i + 1];
        }
// прямой ход метода прогонки
        p[1] = t[1] / m[1];
        q[1] = b[1] / m[1];
        for (int i = 2; i < n; i++) {
            p[i] = t[i] / (m[i] - k[i] * p[i - 1]);
            q[i] = (b[i] + k[i] * q[i - 1]) / (m[i] - k[i] * p[i - 1]);
            System.out.println(p[i] + "\t" + q[i]);
        }
// обратный ход метода прогонки
        y[n] = (b[n] + k[n] * q[n - 1]) / (m[n] - k[n] * p[n - 1]);
        for (int i = n - 1; i >= 1; i--)
            y[i] = p[i] * y[i + 1] + q[i];
        return y;
    }
}
```

Результаты

Вектор найденных значений x:	Вектор значений точного решения:
29.07	29.07
1.97000000000001	1.97
24.65	24.65
-65.850999999999998	-65.85
-37.95	-37.95
-69.41	-69.41
15.88000000000001	15.88
1.23	1.23
21.72	21.72
71.70999999999998	71.71
-97.84	-97.84
95.02000000000002	95.02
6.7909999999999995	6.79
-3.14	-3.14
-3.69	-3.69
-40.46	-40.46
-74.55000000000001	-74.55
-3.32	-3.32
-81.95	-81.95
-90.31999999999998	-90.32
16.51000000000002	16.51