Министерство образования Российской Федерации

Московский государственный технический университет имени Н.Э.Баумана

Факультет информатики и систем управления

Кафедра теоретической информатики и компьютерных технологий

Лабораторная работа №1

“Решение СЛАУ с трехдиагональной матрицей методом прогонки”

по курсу “Численные методы”

Выполнил: Беляев А.В.

Проверила: Домрачева А.Б.

Москва, 2016

Постановка задачи

Найти вектор $\vec{x}$ ($\vec{x} \in R$) в СЛАУ вида $A^3\vec{x} = \vec{d}$ методом прогонки. $A^3 \in R^{n\times n}, \vec{d} \in R$.

Матрица A и вектор $\vec{d}$ заданы.

Основные теоретические сведения

Обозначим числа на главной диагонали через b_i. Числа над главной диагональю через c_i. Числа под главной диагональю через a_i ($i \in 1..n$). Введем нумерацию от левого верхнего к нижнему правому углу. Координаты вектора x обозначим через x_i ($i \in 1..n$).

Перепишем СЛАУ в таком виде:

$$b_1x_1 + c_1x_2 = d_1$$

$$a_1x_1 + b_2x_2 + c_2x_3 = d_2$$

$$\dots$$

$$a_{i-1}x_{i-1} + b_ix_i + c_{i+1}x_{i+1} = d_i$$

$$a_{n-1}x_{n-1} + b_nx_n = d_n$$

Выразим x_1 из первого уравнения:

$$x_1 = \frac{d_1}{b_1} - \frac{c_1}{b_1}x_2$$

Обозначим множитель при x_2 за α_1 и второе слагаемое за β_1:

$$x_1 = \alpha_1x_2 + \beta_1$$

Подставим получившееся выражение во второе уравнение системы:

$$a_1(\alpha_1x_2 + \beta_1) + b_2x_2 + c_2x_3 = d_2$$

Выразим из этого уравнения x_2:

$$x_2 = -\frac{c_2}{a_1\alpha_1 + b_2}x_3 + \frac{d_2 - a_1\beta_1}{a_1\alpha_1 + b_2}$$

Аналогичным образом обозначим множитель перед x_3 за α_2 и второе слагаемое за β_2.

Проведя подобные преобразования, мы можем выразить α_i и β_i. И через них выразим x_i с использованием x_{i+1}:

$$\alpha_i = -\frac{c_i}{a_{i-1}\alpha_{i-1} + b_i}$$

$$\beta_i = -\frac{d_i - a_{i-1}\beta_{i-1}}{a_{i-1}\alpha_{i-1} + b_i}$$

$$x_i = \alpha_i x_{i+1} + \beta_i$$

Выразим значение x_n:

$$x_n = \frac{d_n - a_{n-1}\beta_{n-1}}{a_{n-1}\alpha_{n-1} + b_n}$$

Вычислив подобным образом все значения x_i мы, тем самым, решим систему уравнений.

Практическая реализация работы

В ходе работы была написана программа на языке *C*. На Листинг 1, исходный код представлен исходный код функции программы, основанный на проведенных ранее вычислениях.

Листинг 1, исходный код функции подсчета (для N=4)

```
float x_fl[n];
double x_d[n];

void count_matrix() {
    int i, j;
    _alpha[0] = (-1) * _c[0] / _b[0];
    _beta[0] = _d[0] / _b[0];

    for (i = 1; i < n; i++) {
        _alpha[i] = (-1) * (_c[i]) / (_a[i-1] * _alpha[i-1] + _b[i]);
        _beta[i] = (_d[i] - _a[i-1] * _beta[i-1]) / (_a[i-1] * _alpha[i-1] + _b[i]);
    }

    for (i = 0; i < n; i++) {
        printf("alpha[%d]:%f\n", i, _alpha[i]);
        printf("beta[%d]:%f\n", i, _beta[i]);
```

```
    }

    x_fl[n-1] = _beta[n-1];
    x_d[n-1] = _beta[n-1];

    for (i = n-2; i >= 0; i--) {
        x_fl[i] = _alpha[i] * x_fl[i+1] + _beta[i];
        x_d[i] = _alpha[i] * x_d[i+1] + _beta[i];
    }

    for (i = 0; i < n; i++) {
        printf("float:x[%d]:%f\n", i, x_fl[i]);
        printf("double:x[%d]:%.16f\n", i, x_d[i]);
    }

}
```

Результаты работы

В ходе тестирования на вход программы была подана матрица размера 4 со значениями “4” на главной диагонали и значениями “1” над и под главной диагональю: $A = \begin{pmatrix} 4 & 1 & & \\ 1 & 4 & 1 & \\ & 1 & 4 & 1 \\ & & 1 & 4 \end{pmatrix}$. Вектор $\vec{d}$ задан следующими значениями: $\vec{d} = \begin{pmatrix} 5 \\ 6 \\ 6 \\ 5 \end{pmatrix}$.

В ходе работы программы были вычислены значения x_i вектора x:

$$x = \begin{pmatrix} 1.000000 \\ 1.000000 \\ 1.000000 \\ 1.000000 \end{pmatrix}$$

В данном тесте был получен точный результат.

Но, в проведенных программой вычислениях, ввиду ограничения разрядной сетки и округления, присутствует вычислительная погрешность.

Это видно из полученных значений x_i вектора x на тесте с подобной матрицей *A* и вектором *d*:

$$A = \begin{pmatrix} 23.1 & 12 \\ 9 & 21.6 \end{pmatrix}, \vec{d} = \begin{pmatrix} 47.1 \\ 52.2 \end{pmatrix}.$$

Ожидается результат в виде $x = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$,

но на деле был получен $x = \begin{pmatrix} 1.0000000000000021 \\ 1.9999999999999978 \end{pmatrix}$

Вывод

Описанный выше метод прогонки целесообразно использовать при решении СЛАУ с трехдиагональной матрицей ввиду отсутствия в данном методе методологической погрешности. Однако, при реализации данного метода на ЭВМ возникает вычислительная погрешность.