Методы линейной алгебры

Баев А.Ж.

Содержание

1 Вычисление определителя методом Гаусса

Постановка задачи. Дана квадратная матрица $A \in \mathbb{R}^{n\times n}$. Вычислить определитель $\det A$.

Метод решения. Идея: привести матрицу к треугольному виду элементарными преобразованиями строк (прямой ход метода Гаусса) и найти произведение диагональных элементов.

Выполним n шагов таких, что после k-го шага первые k столбцов будут преобразованы к треугольному виду. На k-м шаге выберем элемент $a_{k,k}$ (ведущий элемент). Далее из всех строк ниже k-й вычтем k-ю строку, умноженную на $\frac{a_{i,k}}{a_{k,k}}$, чтобы обнулить все элементы в столбце под ведущим элементом:

$$a_{i,j} := a_{i,j} - \frac{a_{i,k}}{a_{k,k}} a_{k,j}$$

для всех i от $(k+1)$ до n и для всех j от $(k+1)$ до n.

$$\begin{pmatrix}
a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,k} & a_{1,k+1} & \dots & a_{1,n} \\
0 & a_{2,2} & a_{2,3} & \dots & a_{2,k} & a_{2,k+1} & \dots & a_{2,n} \\
0 & 0 & a_{3,3} & \dots & a_{3,k} & a_{3,k+1} & \dots & a_{3,n} \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
0 & 0 & 0 & \dots & a_{k,k} & a_{k,k+1} & \dots & a_{3,n} \\
0 & 0 & 0 & \dots & 0 & a_{k+1,k+1} & \dots & a_{k+1,n} \\
0 & 0 & 0 & \dots & 0 & a_{k+2,k+1} & \dots & a_{k+2,n} \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
0 & 0 & 0 & \dots & 0 & a_{n,k+1} & \dots & a_{n,n}
\end{pmatrix}$$

Определитель легко найти для треугольной матрицы:

$$\det A = \Pi_{k=1}^{n} a_{k,k}$$

Пример. Приведем к треугольному виду (прямой ход метода Гаусса) и найдем определитель матрицы:

$$\begin{pmatrix}
1 & 1 & 1 & 1 \\
1 & 3 & 2 & 2 \\
2 & 3 & 3 & 3 \\
-1 & 0 & 0 & 1
\end{pmatrix}$$

Шаг 1. Ведущий элемент $a_{1,1} = 1$. Вычтем из 2-й строки 1-ю строку (т.к. $\frac{a_{2,1}}{a_{1,1}} = 1$)

$$\begin{pmatrix}
\boxed{1} & 1 & 1 & 1 \\
\boxed{1} & \boxed{3} & \boxed{2} & \boxed{2} \\
2 & 3 & 3 & 3 \\
-1 & 0 & 0 & 1
\end{pmatrix} \Rightarrow \begin{pmatrix}
1 & 1 & 1 & 1 \\
0 & 2 & 1 & 1 \\
2 & 3 & 3 & 3 \\
-1 & 0 & 0 & 1
\end{pmatrix}$$

Вычтем из 3-й строки 1-ю строку, умноженную на 2 (т.к. $\frac{a_{3,1}}{a_{1,1}} = 2$)

$$\begin{pmatrix} \boxed{1} & 1 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ \boxed{2} & \boxed{3} & \boxed{3} & \boxed{3} \\ -1 & 0 & 0 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ -1 & 0 & 0 & 1 \end{pmatrix}$$

Вычтем из 4-й строки 1-ю строку, умноженную на -1 (т.к. $\frac{a_{4,1}}{a_{1,1}} = -1$)

$$\begin{pmatrix} \boxed{1} & 1 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ \boxed{-1} & \boxed{0} & \boxed{0} & \boxed{1} \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 2 \end{pmatrix}$$

Шаг 2. Ведущий элемент $a_{2,2} = 2$. Вычтем из 3-й строки 2-ю строку, умноженную на $\frac{1}{2}$ (т.к. $\frac{a_{3,2}}{a_{2,2}} = \frac{1}{2}$)

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & \boxed{2} & 1 & 1 \\ 0 & \boxed{1} & \boxed{1} & \boxed{1} \\ 0 & 1 & 1 & 2 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & 1 & 2 \end{pmatrix}$$

Вычтем из 4-й строки 2-ю строку, умноженную на $\frac{1}{2}$ (т.к. $\frac{a_{4,2}}{a_{2,2}} = \frac{1}{2}$)

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & \boxed{2} & 1 & 1 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & \boxed{1} & \boxed{1} & \boxed{2} \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{3}{2} \end{pmatrix}$$

Шаг 3. Ведущий элемент $a_{3,3} = \frac{1}{2}$. Вычтем из 4-й строки 3-ю строку (т.к. $\frac{a_{4,3}}{a_{3,3}} = 1$)

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & \boxed{\frac{1}{2}} & \frac{1}{2} \\ 0 & 0 & \boxed{\frac{1}{2}} & \boxed{\frac{3}{2}} \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Определитель треугольной матрицы равен произведению диагональных элементов. Соответственно: $\det A = 1 * 2 * \frac{1}{2} * 1 = 1$.

Псевдокод.

```
for k := 1 .. n
    for i := k+1 .. n
        t := A[i][k] / A[k][k]
        A[i][k] := 0
        for j := k+1 .. n
            A[i][j] := A[i][j] - t * A[k][j]

det := 1
for k := 1 .. n
    det := det * A[k][k]
```

Алгоритмическая сложность. Посчитаем алгоритмическую сложность:

1. прямого хода метода Гаусса;
2. вычисления определителя треугольной матрицы;
3. вычисления определителя произвольной матрицы.

1. Алгоритмическая сложность прямого хода метода Гаусса. Наиболее весомые операции в циклах – это умножение и деление. Считаем количество умножений и делений в первой части алгоритма:

$$g(n) = \sum_{k=1}^{n} \sum_{i=k+1}^{n} \left(1 + \sum_{j=k+1}^{n} 1 \right) = \sum_{k=1}^{n} \sum_{i=k+1}^{n} (1 + n - k) = \sum_{k=1}^{n} (n-k)(n+1-k)$$

Сделаем замену $k = n + 1 - s$:

$$g(n) = \sum_{s=1}^{n} (s-1)s = \sum_{s=1}^{n} s^2 - \sum_{s=1}^{n} s = \frac{n(n+1)(2n+1)}{6} - \frac{n(n+1)}{2} = \frac{n^3 - n}{3}$$

2. Алгоритмическая сложность вычисления определителя треугольной матрицы (без прямого хода метода Гаусса):

$$d(n) = \sum_{k=1}^{n} 1 = n$$

.

3. Алгоритмическую сложность вычисления определителя произвольной матрицы:

$$f(n) = g(n) + d(n) = \frac{n^3 + 2n}{3}$$

Асимптотика алгоритмической сложности:

1. прямого хода метода Гаусса $g(n) = \Theta(n^3)$;
2. вычисления определителя треугольной матрицы $d(n) = \Theta(n)$;
3. вычисления определителя произвольной матрицы $f(n) = \Theta(n^3)$.

2 Решение системы уравнений методом Гаусса

Постановка задачи. Решить систему линейный алгебраических уравнений:

$$Ax = f$$

где $A \in \mathbb{R}^{n \times n}$, $x \in \mathbb{R}^n$, $f \in \mathbb{R}^n$.

Метод решения. Идея: привести матрицу к треугольному виду элементарными преобразованиями строк (прямой ход метода Гаусса) и найти решения начиная с последней компененты (обратный ход метода Гаусса).

Прямой ход метода Гаусса. Выполним n шагов таких, что после k-го шага первые k столбцов будут преобразованы к треугольному виду:

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,k} & a_{1,k+1} & \dots & a_{1,n} \\ 0 & a_{2,2} & a_{2,3} & \dots & a_{2,k} & a_{2,k+1} & \dots & a_{2,n} \\ 0 & 0 & a_{3,3} & \dots & a_{3,k} & a_{3,k+1} & \dots & a_{3,n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & a_{k,k} & a_{k,k+1} & \dots & a_{3,n} \\ 0 & 0 & 0 & \dots & 0 & a_{k+1,k+1} & \dots & a_{k+1,n} \\ 0 & 0 & 0 & \dots & 0 & a_{k+2,k+1} & \dots & a_{k+2,n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & a_{n,k+1} & \dots & a_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_k \\ x_{k+1} \\ x_{k+2} \\ \dots \\ x_n \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \dots \\ f_k \\ f_{k+1} \\ f_{k+2} \\ \dots \\ f_n \end{pmatrix}$$

На k-м шаге выберем элемент $a_{k,k}$ (ведущий элемент). Далее из всех строк ниже k-й вычтем k-ю строку, умноженную на $\frac{a_{i,k}}{a_{k,k}}$, чтобы обнулить все элементы в столбце под ведущим элементом:

$$a_{i,j} := a_{i,j} - \frac{a_{i,k}}{a_{k,k}} a_{k,j}$$

для всех i от $(k+1)$ до n и для всех j от $(k+1)$ до n. И таким же образом преобразовать правую часть:

$$f_i := f_i - \frac{a_{i,k}}{a_{k,k}} f_k$$

.

Обратный ход метода Гаусса. Чтобы выразить решения, выпишем f_i из соотношения $Ax = f$, где A – верхнетреугольная матрица.

$$\sum_{j=i}^{n} a_{i,j} x_j = f_i$$

для всех i от n до 1.

Выразим x_i:

$$x_i = \frac{f_i - \sum_{j=i+1}^{n} a_{i,j} x_j}{a_{i,i}}$$

для всех i от n до 1.

Пример. Приведем к треугольному виду (прямой ход метода Гаусса) и найдем решения (обратный ход метода Гаусса):

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 3 & 2 & 2 \\ 2 & 3 & 3 & 3 \\ -1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 4 \\ -3 \end{pmatrix}$$

Приведение к треугольному виду:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 3 & 2 & 2 \\ 2 & 3 & 3 & 3 \\ -1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 4 \\ -3 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 0 \\ -1 \end{pmatrix} \Rightarrow$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{3}{2} \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ -\frac{1}{2} \\ -\frac{3}{2} \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ -\frac{1}{2} \\ -1 \end{pmatrix} \Rightarrow$$

Обратный ход.

$$x_4 = \frac{f_4}{a_{4,4}} = \frac{-1}{1} = -1$$

$$x_3 = \frac{f_3 - a_{3,4}x_4}{a_{3,3}} = \frac{-\frac{1}{2} - \frac{1}{2} * (-1)}{\frac{1}{2}} = 0$$

$$x_2 = \frac{f_2 - a_{2,3}x_3 - a_{2,4}x_4}{a_{2,2}} = \frac{1 - 1 * 0 - 1 * (-1)}{2} = 1$$

$$x_1 = \frac{f_1 - a_{1,2}x_2 - a_{1,3}x_3 - a_{1,4}x_4}{a_{1,1}} = \frac{2 - 1 * 1 - 1 * 0 - 1 * (-1)}{1} = 2$$

Ответ: $x = (2, 1, 0, -1)^T$.

Псевдокод.

```
for k := 1 .. n
    for i := k+1 .. n
        t := A[i][k] / A[k][k]
        A[i][k] := 0
        for j := k+1 .. n
            A[i][j] := A[i][j] - t * A[k][j]
        f[i] := f[i] - t * f[k]

for i := n .. 1
    x[i] := f[i]
    for j := i+1 .. n
        x[i] := x[i] - A[i][j] * x[j]
    x[i] := x[i] / A[i][i]
```

Алгоритмическая сложность. Посчитаем алгоритмическую сложность:

1. прямого хода метода Гаусса;
2. обратного хода метода Гаусса;
3. метода Гаусса для решения СЛАУ.

1. Алгоритмическая сложность прямого хода метода Гаусса. Наиболее весомые операции в циклах – это умножение и деление. Считаем количество умножений и делений в первой части алгоритма:

$$f_1(n) = \sum_{k=1}^{n} \sum_{i=k+1}^{n} \left(2 + \sum_{j=k+1}^{n} 1 \right) = \sum_{k=1}^{n} \sum_{i=k+1}^{n} (2+n-k) = \sum_{k=1}^{n} (n-k)(n+2-k)$$

Сделаем замену $k = n + 1 - s$:

$$f_1(n) = \sum_{s=1}^{n} (s-1)(s+1) = \sum_{s=1}^{n} s^2 - \sum_{s=1}^{n} 1 =$$

$$= \frac{n(n+1)(2n+1)}{6} - n = \frac{2n^3 + 3n^2 - 5n}{6}$$

2. Алгоритмическая сложность обратного хода метода Гаусса. Наиболее весомые операции в циклах – это умножение и деление. Считаем количество умножений и делений во второй части алгоритма:

$$f_2(n) = \sum_{i=1}^{n} \left(1 + \sum_{j=i+1}^{n} 1 \right) = \sum_{i=1}^{n} (1 + n - i)$$

Сделаем замену $i = n + 1 - s$:

$$f_2(n) = \sum_{s=1}^{n} s = \frac{n^2 + n}{2}$$

3. Алгоритмическая сложность метода Гаусса для решения СЛАУ:

$$f(n) = f_1(n) + f_2(n) = \frac{2n^3 + 3n^2 - 5n}{6} + \frac{n^2 + n}{2} = \frac{n^3 + 3n^2 - n}{3}$$

Асимптотика алгоритмической сложности:

1. прямого хода метода Гаусса $f_1(n) = \Theta(n^3)$;
2. обратного хода метода Гаусса $f_2(n) = \Theta(n^2)$;
3. метода Гаусса для решения СЛАУ $f(n) = \Theta(n^3)$.

3 Нахождение обратной матрицы

Постановка задачи. Дана квадратная невырожденная квадратная матрица $A \in \mathbb{R}^{n\times n}$. Вычислить обратную матрицу A^{-1}.

Метод решения. Идея: решить матричное уравнени $AX = E$.

$$\begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{pmatrix} \begin{pmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \dots & \dots & \dots & \dots \\ x_{n,1} & x_{1,n} & \dots & x_{n,n} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

Для этого приведем матрицу A к треугольному виду (вместе с правой частью, которая изначально является единичной матрицей) прямым ходом метода Гаусса:

$$\begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ 0 & a_{2,2} & \dots & a_{2,n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{n,n} \end{pmatrix} \begin{pmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \dots & \dots & \dots & \dots \\ x_{n,1} & x_{1,n} & \dots & x_{n,n} \end{pmatrix} =$$

$$= \begin{pmatrix} f_{1,1} & f_{1,2} & \dots & f_{1,n} \\ f_{2,1} & f_{2,2} & \dots & f_{2,n} \\ \dots & \dots & \dots & \dots \\ f_{n,1} & f_{n,2} & \dots & f_{n,n} \end{pmatrix}$$

Далее для каждого столбца матрицы X решим СЛАУ с треугольной матрицей обратным методом Гаусса.

Прямой ход:

$$a_{i,j} := a_{i,j} - \frac{a_{i,k}}{a_{k,k}} a_{k,j}$$

для всех i от $(k+1)$ до n и для всех j от $(k+1)$ до n.

$$f_{i,j} := f_{i,j} - \frac{a_{i,k}}{a_{k,k}} f_{k,j}$$

для всех i от $(k+1)$ до n и для всех j от 1 до k (для остальных j вычитаемое будет нулем).

Обратный ход:

$$x_{i,j} = \frac{f_{i,j} - \sum_{k=i+1}^{n} a_{i,k} x_{k,j}}{a_{i,i}}$$

для всех i от n до 1 и для всех j от 1 до n.

Пример. Прямой ход метода Гаусса:

$$\left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 3 & 2 & 2 & 0 & 1 & 0 & 0 \\ 2 & 3 & 3 & 3 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right)$$

$$\left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 1 & -1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & -2 & 0 & 1 & 0 \\ 0 & 1 & 1 & 2 & 1 & 0 & 0 & 1 \end{array}\right)$$

$$\left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & -\frac{3}{2} & -\frac{1}{2} & 1 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{3}{2} & \frac{3}{2} & -\frac{1}{2} & 0 & 1 \end{array}\right)$$

$$\left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & -\frac{3}{2} & -\frac{1}{2} & 1 & 0 \\ 0 & 0 & 0 & 1 & 3 & 0 & -1 & 1 \end{array}\right)$$

Обратый ход метода Гаусса для первого столбца:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{1,1} \\ x_{2,1} \\ x_{3,1} \\ x_{4,1} \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ -\frac{3}{2} \\ 3 \end{pmatrix}$$

Откуда легко найти: $(x_{1,1}, x_{2,1}, x_{3,1}, x_{4,1}) = (3, 1, -6, 3)$.

Обратый ход метода Гаусса для второго столбца:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{1,2} \\ x_{2,2} \\ x_{3,2} \\ x_{4,2} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ -\frac{1}{2} \\ 0 \end{pmatrix}$$

Откуда легко найти: $(x_{1,2}, x_{2,2}, x_{3,2}, x_{4,2}) = (0, 1, -1, 0)$.

Обратый ход метода Гаусса для третьего столбца:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{1,3} \\ x_{2,3} \\ x_{3,3} \\ x_{4,3} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 \end{pmatrix}$$

Откуда легко найти: $(x_{1,3}, x_{2,3}, x_{3,3}, x_{4,3}) = (-1, -1, 3, -1)$.

Обратый ход метода Гаусса для третьего столбца:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{1,4} \\ x_{2,4} \\ x_{3,4} \\ x_{4,4} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Откуда легко найти: $(x_{1,4}, x_{2,4}, x_{3,4}, x_{4,4}) = (0, 0, -1, 1)$.

Обратная матрица:

$$A^{-1} = \begin{pmatrix} 3 & 0 & -1 & 0 \\ 1 & 1 & -1 & 0 \\ -6 & -1 & 3 & -1 \\ 3 & 0 & -1 & 1 \end{pmatrix}$$

Псевдокод. F – единичная матрица, A – данная матрица, X – матрица для ответа.

```
for k := 1 .. n
    for i := k+1 .. n
        t := A[i][k] / A[k][k]
        A[i][k] := 0
        for j := k+1 .. n
            A[i][j] := A[i][j] - t * A[k][j]
        for j := 1 .. k
            F[i][j] := F[i][j] - t * F[k][j]

for j := 1 .. n
    for i := n .. 1
        X[i][j] := F[i][j]
        for k := i+1 .. n
            X[i][j] := X[i][j] - A[i][k] * X[k][j]
        X[i][j] := X[i][j] / A[i][i]
```

При реализации можно использовать матрицу F для хранения значений матрицы X, так как при рассчете $X_{i,j}$ достаточно знать только $F_{i,j}$.

Алгоритмическая сложность. Посчитаем алгоритмическую сложность.

1. Алгоритмическая сложность прямого хода метода Гаусса. Наиболее весомые операции в циклах – это умножение и деление. Считаем количество умножений и делений в первой части алгоритма:

$$f_1(n) = \sum_{k=1}^{n} \sum_{i=k+1}^{n} \left(1 + \sum_{j=k+1}^{n} 1 + \sum_{j=1}^{k} 1 \right) = \sum_{k=1}^{n} \sum_{i=k+1}^{n} (1+n) =$$

$$= (n+1) \sum_{k=1}^{n} (n-k) = (n+1) \sum_{s=1}^{n-1} s = \frac{n(n^2-1)}{2} = \frac{n^3-n}{2}$$

2. Алгоритмическая сложность обратного хода метода Гаусса. Наиболее весомые операции в циклах – это умножение и деление. Считаем количество умножений и делений во второй части алгоритма:

$$f_2(n) = \sum_{j=1}^{n} \sum_{i=1}^{n} \left(1 + \sum_{k=i+1}^{n} 1 \right) = \sum_{j=1}^{n} \sum_{i=1}^{n} (1+n-i) = \sum_{j=1}^{n} \sum_{s=1}^{n} s =$$

$$= \frac{n^2(n+1)}{2} = \frac{n^3+n}{2}$$

3. Алгоритмическая сложность метода Гаусса для решения СЛАУ:

$$f(n) = f_1(n) + f_2(n) = n^3$$

4 Решение системы уравнений итерационными методами Зейделя и Якоби

Постановка задачи. Решить систему линейный алгебраических уравнений:

$$Ax = f$$

где $A \in \mathbb{R}^{n \times n}$, $x \in \mathbb{R}^n$, $f \in \mathbb{R}^n$.

Метод решения. Идея: перепишем задачу в эквивалентной форме

$$x = G(x)$$

Далее задаем начальное приближение x_0, вычисляем $x_{k+1} = G(x_k)$ итерационно до тех пор, пока вектора x_k и x_{k+1} не станут достаточно близкими. Данных подход называет методом простой итерации.

В частности, представим $A = L + D + U$, где L - нижнетреугольная, D - диагональная, U - верхнетреугольная матрица.

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n-1} & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} & a_{2,n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n-1,1} & a_{n-1,2} & \dots & a_{n-1,n-1} & a_{n-1,n} \\ a_{n,1} & a_{n,2} & \dots & a_{n,n-1} & a_{n,n} \end{pmatrix}$$

$$L = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 \\ a_{2,1} & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ a_{n-1,1} & a_{n-1,2} & \dots & 0 & 0 \\ a_{n,1} & a_{n,2} & \dots & a_{n,n-1} & 0 \end{pmatrix}$$

$$D = \begin{pmatrix} a_{1,1} & 0 & \dots & 0 & 0 \\ 0 & a_{2,2} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{n-1,n-1} & 0 \\ 0 & 0 & \dots & 0 & a_{n,n} \end{pmatrix}$$

$$U = \begin{pmatrix} 0 & a_{1,2} & \dots & a_{1,n-1} & a_{1,n} \\ 0 & 0 & \dots & a_{2,n} & a_{2,n} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & a_{n-1,n} \\ 0 & 0 & \dots & 0 & 0 \end{pmatrix}$$

Запишем уравнение $Lx+Dx+Ux = f$. Определим некоторые слагаемые как вектор на k-й итерации, а остальные на $(k+1)$-й итерации.

Метод Якоби:

$$Lx^k + Dx^{k+1} + Ux^k = f$$

Выпишем поэлементно:

$$\sum_{j=1}^{i-1} a_{i,j}x_j^k + a_{i,i}x_i^{k+1} + \sum_{j=i+1}^{n} a_{i,j}x_j^k = f_i$$

Выразим x^{k+1}:

$$x_i^{k+1} = \frac{1}{a_{i,i}}\left(f_i - \sum_{j=1}^{i-1} a_{i,j}x_j^k - \sum_{j=i+1}^{n} a_{i,j}x_j^k\right)$$

Заметим, что при реализации нет необходимо хранить все слои, кроме последнего. Далее значения на новом слое будет обозначен $\tilde{x}$, а на старом x.

Метод Зейделя:

$$Lx^{k+1} + Dx^{k+1} + Ux^k = f$$

Выпишем поэлементно:

$$\sum_{j=1}^{i-1} a_{i,j}x_j^{k+1} + a_{i,i}x_i^{k+1} + \sum_{j=i+1}^{n} a_{i,j}x_j^k = f_i$$

Выразим x^{k+1}:

$$x_i^{k+1} = \frac{1}{a_{i,i}}\left(f_i - \sum_{j=1}^{i-1} a_{i,j}x_j^{k+1} - \sum_{j=i+1}^{n} a_{i,j}x_j^k\right)$$

Заметим, что при реализации нет необходимо хранить все слои, кроме последнего. Далее значения на новом слое будет обозначен $\tilde{x}$, а на старом x.

Пример. Решим методом Якоби:

$$\begin{pmatrix} 2 & -1 & 0 & -1 \\ 0 & 2 & -1 & 0 \\ -1 & 1 & 3 & -0 \\ 1 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 4 \\ 3 \\ 2 \\ 1 \end{pmatrix}$$

Представим A как сумму диагональной и треугольных матриц:

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & 0 & -2 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix} \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \\ \tilde{x}_4 \end{pmatrix} +$$

$$+ \begin{pmatrix} 0 & -1 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 4 \\ 3 \\ 2 \\ 1 \end{pmatrix}$$

Выразим явно:

$$\tilde{x}_1 = \frac{4 - (-1) * x_2 - 0 * x_3 - (-1) * x_4}{2}$$
$$\tilde{x}_2 = \frac{3 - 0 * x_1 - (-1) * x_3 - 0 * x_4}{2}$$
$$\tilde{x}_3 = \frac{2 - (-1) * x_1 - 1 * x_2 - 0 * x_4}{3}$$
$$\tilde{x}_4 = \frac{1 - 1 * x_1 - 0 * x_2 - (-2) * x_3}{4}$$

В качестве начального приближения выберем нулевой вектор, а допустимую точность выберем $\varepsilon = 0.2$ Выпишем итерации:

$$(0,0,0,0) \to \left(2, \frac{3}{2}, \frac{2}{3}, \frac{1}{4}\right) \to \left(\frac{23}{8}, \frac{11}{6}, \frac{5}{6}, \frac{1}{12}\right) \to \left(\frac{71}{24}, \frac{23}{12}, \frac{73}{72}, -\frac{5}{96}\right)$$

Разница между последними двумя векторами $||x - \tilde{x}||^2 < \varepsilon^2$. Ответ: $x = (2.95833, 1.91667, 1.01389, -0.052083)^T$ (для сравнения, точное решение $(3, 2, 1, 0)$).

Решим эту же задачу методом Зейделя. Представим A как сумму диагональной и треугольных матриц:

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & 0 & -2 & 0 \end{pmatrix} \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \\ \tilde{x}_4 \end{pmatrix} + \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix} \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \\ \tilde{x}_4 \end{pmatrix} +$$
$$+ \begin{pmatrix} 0 & -1 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 4 \\ 3 \\ 2 \\ 1 \end{pmatrix}$$

Выразим явно:

$$\tilde{x}_1 = \frac{4 - (-1) * x_2 - 0 * x_3 - (-1) * x_4}{2}$$
$$\tilde{x}_2 = \frac{3 - 0 * \tilde{x}_1 - (-1) * x_3 - 0 * x_4}{2}$$
$$\tilde{x}_3 = \frac{2 - (-1) * \tilde{x}_1 - 1 * \tilde{x}_2 - 0 * x_4}{3}$$
$$\tilde{x}_4 = \frac{1 - 1 * \tilde{x}_1 - 0 * x_2 - (-2) * x_3}{4}$$

В качестве начального приближения выберем нулевой вектор, а допустимую точность выберем $\varepsilon = 0.2$ Выпишем итерации:

$$(0,0,0,0) \to \left(2, \frac{3}{2}, \frac{5}{6}, \frac{1}{6}\right) \to \left(\frac{17}{6}, \frac{23}{12}, \frac{35}{36}, \frac{1}{36}\right)$$

Разница между последними двумя векторами $||x - \tilde{x}||^2 < \varepsilon^2$. Ответ: $x = (2.83333, 1.91667, 0.972222, 0.027778)^T$ (для сравнения, точное решение $(3, 2, 1, 0)$).

Псевдокод. Метод простой итерации в общем виде:

```
x2 := 0
do
    for i := 1 .. n
        x1[i] := x2[i]
    G(n, x1, x2)
while Diff(n, x1, x2) > eps2
```

Функция $Diff(n, x1, x2)$ реализует вычисление квадрата нормы разности (обратите внимание, что можно не извлекать корень, если квадрат нормы сравнивать с предаврительно вычисленным числом $eps2 = \varepsilon^2$):

```
diff(n, x1, x2)
    s := 0
    for i := 1 .. n
        s := s + (x2[i] - x1[i]) * (x2[i] - x1[i])
    return s
```

Процедура $G(n, x1, x2)$ реализует вычисления нового слоя. В случае метода Якоби:

```
jacobi(n, x1, x2)
    for i := 1 .. n
        s := 0
        for j := 1 .. i-1
            s := s + A[i][j] * x1[j]
        for j := i+1 .. n
            s := s + A[i][j] * x1[j]
        x2[i] := (f[i] - s) / A[i][i]
    return
```

В случае метода Зейделя:

```
seidel(n, x1, x2)
    for i := 1 .. n
        s := 0
        for j := 1 .. i-1
            s := s + A[i][j] * x2[j]
        for j := i+1 .. n
            s := s + A[i][j] * x1[j]
        x2[i] := (f[i] - s) / A[i][i]
    return
```

Алгоритмическая сложность. Посчитаем алгоритмическую сложность:

1. вычисления нормы разности;
2. итерационного шага метода Якоби, Зейделя;
3. метода простой итерации.

1. Алгоритмическая сложность вычисление нормы разности:

$$d(n) = n$$

2. Алгоритмическая сложность итерационного шага метода Якоби (Зейделя):

$$g(n) = \sum_{i=1}^{n} \left(1 + \sum_{j=1}^{i+1} 1 + \sum_{j=i+1}^{n} 1 \right) = \sum_{i=1}^{n} n = n^2$$

3. Алгоритмическая сложность метода простой итерации:

$$f(n) = M(\varepsilon, A, x_0) * (g(n) + d(n)) = M(\varepsilon, A, x_0) * (n^2 + n)$$

Асимптотика алгоритмической сложности достаточно сложна и сильно зависит от начального приближения и свойств исходной матрицы.

5 Метод прогонки

Постановка задачи. Дана квадратная трехдиагональная матрица $A \in \mathbb{R}^{n\times n}$. Решить СЛАУ $Ax = f$.

Метод решения. Идея: $x_{i-1} = \alpha_i x_i + \beta_i$.

$$\begin{pmatrix} b_1 & c_1 & 0 & 0 & \dots & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & \dots & 0 & 0 & 0 \\ 0 & a_3 & b_3 & c_3 & \dots & 0 & 0 & 0 \\ 0 & 0 & a_4 & b_4 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & b_{n-2} & c_{n-2} & 0 \\ 0 & 0 & 0 & 0 & \dots & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & 0 & 0 & 0 & \dots & 0 & a_n & b_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ \vdots \\ f_{n-2} \\ f_{n-1} \\ f_n \end{pmatrix}$$

Перепишем СЛАУ в виде построчных соотношений:

$$\begin{cases} b_1x_1 + c_1x_2 & = f_1 \\ a_ix_{i-1} + b_ix_i + c_ix_{i+1} & = f_i, i = 2, ..., n-1 \\ a_nx_{n-1} + b_nx_n & = f_n \end{cases} \quad (1)$$

Добавим фиктивные элементы и корни: $a_1 = c_n = 0$ и $x_0 = x_{n+1} = 0$:

$$a_ix_{i-1} + b_ix_i + c_ix_{i+1} = f_i, i = 1, ..., n$$

Пусть элементы решения выражаются линейно друг через друга :

$$x_{i-1} = \alpha_i x_i + \beta_i, i = 1, ..., n+1 \quad (2)$$

Подставим (2) в соотношение (1) при $i = 1, ..., n$:

$$a_i(\alpha_i x_i + \beta_i) + b_ix_i + c_ix_{i+1} = f_i$$

$$(a_i\alpha_i + b_i)x_i + c_ix_{i+1} = f_i - a_i\beta_i$$

$$x_i = \frac{-c_i}{a_i\alpha_i + b_i}x_{i+1} + \frac{f_i - a_i\beta_i}{a_i\alpha_i + b_i}$$

Из последнего соотношения получаем:

$$\begin{cases} \alpha_{i+1} = \dfrac{-c_i}{a_i\alpha_i + b_i} \\ \beta_{i+1} = \dfrac{f_i - a_i\beta_i}{a_i\alpha_i + b_i} \end{cases} i = 1, ..., n$$

с учетом того, что $x_0 = 0$ получаем $\alpha_1 = \beta_1 = 0$.

Прямой ход прогонки:

$$\begin{cases} \alpha_1 = 0, \beta_1 = 0, \\ \alpha_{i+1} = \dfrac{-c_i}{a_i\alpha_i + b_i}, i = 1, ..., n \\ \beta_{i+1} = \dfrac{f_i - a_i\beta_i}{a_i\alpha_i + b_i}, i = 1, ..., n \end{cases} \tag{3}$$

Схема вычисления:

$$\begin{tikzcd}
\alpha_1 = 0 \arrow[r] \arrow[dr] & \alpha_2 \arrow[r] \arrow[dr] & \alpha_3 \arrow[r] \arrow[dr] & ... \arrow[r] \arrow[dr] & \alpha_n \arrow[r] \arrow[dr] & \alpha_{n+1} \\
\beta_1 = 0 \arrow[r] & \beta_2 \arrow[r] & \beta_3 \arrow[r] & ... \arrow[r] & \beta_n \arrow[r] & \beta_{n+1}
\end{tikzcd}$$

Обратный ход прогонки:

$$\begin{cases} x_{n+1} = 0 \\ \quad x_i = \alpha_{i+1}x_{i+1} + \beta_{i+1}, i = \overline{n, 1} \end{cases} \tag{4}$$

Схема вычисления:

$$\begin{tikzcd}
\alpha_1 & \alpha_2 \arrow[dl] & \alpha_3 \arrow[dl] & ... \arrow[dl] & \alpha_{n-1} \arrow[dl] & \alpha_n \arrow[dl] & \alpha_{n+1} \arrow[dl] \\
x_1 & x_2 \arrow[l] & x_3 \arrow[l] & ... \arrow[l] & x_{n-1} \arrow[l] & x_n \arrow[l] & x_{n+1} = 0 \arrow[l] \\
\beta_1 & \beta_2 \arrow[ul] & \beta_3 \arrow[ul] & ... \arrow[ul] & \beta_{n-1} \arrow[ul] & \beta_n \arrow[ul] & \beta_{n+1} \arrow[ul]
\end{tikzcd}$$

Пример.

$$\begin{pmatrix} 4 & 3 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 10 \\ 10 \\ 8 \end{pmatrix}$$

Запишем вектора a, b, c и f:

$$a = (0, 1, 1)$$

$$b = (4, 3, 2)$$

$$c = (3, 1, 0)$$

$$f = (10, 10, 8)$$

Вычислим коэффициенты α и β прямым ходом прогонки (3):

$\alpha_1 = 0$ $\qquad$ $\beta_1 = 0$

$\alpha_2 = \frac{-c_1}{a_1\alpha_1+b_1} = \frac{-3}{0*0+4} = -\frac{3}{4}$ $\qquad$ $\beta_2 = \frac{f_1-a_1\beta_1}{a_1\alpha_1+b_1} = \frac{10-0*0}{0*0+4} = \frac{5}{2}$

$\alpha_3 = \frac{-c_2}{a_2\alpha_2+b_2} = \frac{-1}{1*(-\frac{3}{4})+3} = -\frac{4}{9}$ $\qquad$ $\beta_3 = \frac{f_2-a_2\beta_2}{a_2\alpha_2+b_2} = \frac{10-1*\frac{5}{2}}{1*(-\frac{3}{4})+3} = \frac{10}{3}$

$\alpha_4 = \frac{-c_3}{a_3\alpha_3+b_3} = \frac{0}{1*(-\frac{4}{9})+2} = 0$ $\qquad$ $\beta_4 = \frac{f_3-a_3\beta_3}{a_3\alpha_3+b_3} = \frac{8-1*\frac{10}{3}}{1*(-\frac{4}{9})+2} = 3$

Вычислим решение x обратным ходом прогонки (4):
$x_4 = 0$
$x_3 = \alpha_4 x_4 + \beta_4 = 0*0+3 = 3$
$x_2 = \alpha_3 x_3 + \beta_3 = -\frac{4}{9}*3+\frac{10}{3} = 2$
$x_1 = \alpha_2 x_2 + \beta_2 = -\frac{3}{4}*2+\frac{5}{2} = 1$
Ответ: $x = (1, 2, 3)$ (фиктивный элемент x_4 в ответ не включается).

Псевдокод.

```
a[1] := 0
c[n] := 0
alpha[1] := 0
beta[1] := 0
for i := 1 .. n
    d := a[i] * alpha[i] + b[i]
    alpha[i+1] := - c[i] / d
    beta[i+1] := (f[i] - a[i] * beta[i]) / d
x[n+1] := 0
for i := n .. 1
    x[i] := alpha[i+1] * x[i+1] + beta[i+1]
```

Алгоритмическая сложность. Посчитаем алгоритмическую сложность:

1. прямого хода метода прогонки — $f(n) = 4n$;
2. обратного хода метода прогонки — $f(n) = n$;
3. всего метода прогонки — $f(n) = 5n$.

Асимптотика прямого и обратного хода: $\Theta(n)$.