

На первом шаге алгоритма Гаусса выберем диагональный элемент  $a_{11} \neq 0$  (если он равен 0, то первую строку переставляем с какой-либо нижележащей строкой) и объявляем

его *ведущим*, а соответствующую строку и столбец, на пересечении которых он стоит - *ведущими*. Обнулим элементы  $a_{21}, \dots, a_{n1}$  ведущего столбца. Для этого сформируем числа  $(-a_{21}/a_{11}), (-a_{31}/a_{11}), \dots, (-a_{n1}/a_{11})$ . Умножая ведущую строку на число  $(-a_{21}/a_{11})$ , складывая со второй и ставя результат на место второй строки, получим вместо элемента  $a_{21}$  нуль, а вместо элементов  $a_{2j}, j = \overline{2, n}$ ,  $b_2$  - соответственно элементы  $a_{2j}^1 = a_{2j} + a_{1j}(-a_{21}/a_{11})$ ,  $j = \overline{2, n}$ ,  $b_2^1 = b_2 + b_1(-a_{21}/a_{11})$  и т.д. Умножая ведущую строку на число  $(-a_{n1}/a_{11})$ , складывая с  $n$ -ой строкой и ставя результат на место  $n$ -ой строки, получим вместо элемента  $a_{n1}$  нуль, а остальные элементы этой строки будут иметь вид:  $a_{nj}^1 = a_{nj} + a_{1j}(-a_{n1}/a_{11})$ ,  $b_n^1 = b_n + b_1(-a_{n1}/a_{11})$ . Сохраняя ведущую строку неизменной, получим в результате 1-го шага алгоритма Гаусса следующую матрицу:

$$\text{Ведущая строка} \rightarrow \begin{array}{c|cccc|c} x_1 & x_2 & x_3 & \cdots & x_n & b \\ \hline a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ \hline 0 & a_{22}^1 & a_{23}^1 & \cdots & a_{2n}^1 & b_2^1 \\ \hline 0 & a_{32}^1 & a_{33}^1 & \cdots & a_{3n}^1 & b_3^1 \\ \hline \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \hline 0 & a_{n2}^1 & a_{n3}^1 & \cdots & a_{nn}^1 & b_n^1 \end{array} \xrightarrow[2\text{-й шаг}]{\left(-\frac{a_{32}^1}{a_{22}^1}\right), \dots, \left(-\frac{a_{n2}^1}{a_{22}^1}\right)} \begin{array}{c} \uparrow \text{Ведущий столбец} \end{array}$$

На втором шаге алгоритма Гаусса в качестве ведущего элемента выбирается элемент  $a_{22}^1 \neq 0$  (если он равен нулю, то вторую строку взаимно меняем на *нижележащую* строку).

Формируются числа  $\left(-\frac{a_{32}^1}{a_{22}^1}\right), \dots, \left(-\frac{a_{n2}^1}{a_{22}^1}\right)$ , которые ставятся около ведущей строки.

Умножая ведущую строку на число  $\left(-\frac{a_{32}^1}{a_{22}^1}\right)$  и складывая результат с третьей строкой,

получим вместо элемента  $a_{32}^1$  нуль, а вместо элементов  $a_{3j}^1, j = \overline{3, n}$ ,  $b_3^1, \Sigma_3^1$  - элементы

$a_{3j}^2 = a_{3j}^1 + a_{2j}^1 \left(-\frac{a_{32}^1}{a_{22}^1}\right)$ ,  $j = \overline{3, n}$ ,  $b_3^2 = b_3^1 + b_2^1 \left(-\frac{a_{32}^1}{a_{22}^1}\right)$ ,. И так далее. Умножая ведущую

строку на число  $\left(-\frac{a_{n2}^1}{a_{22}^1}\right)$ , складывая результат с  $n$ -ой строкой и ставя полученную сумму

на место  $n$ -ой строки, получим вместо элемента  $a_{n2}^1$  нуль, а вместо элементов  $a_{nj}^1, b_n^1, \Sigma_n^1$

- элементы  $a_{nj}^2 = a_{nj}^1 + a_{2j}^1 \left(-\frac{a_{n2}^1}{a_{22}^1}\right)$ ,  $j = \overline{3, n}$ ,  $b_n^2 = b_n^1 + b_2^1 \left(-\frac{a_{n2}^1}{a_{22}^1}\right)$ . Сохраняя 1-ую и 2-ую

$$\begin{array}{c} \begin{array}{cccccc} x_1 & x_2 & x_3 & \cdots & x_n & b \end{array} \\ \left[ \begin{array}{ccc|ccc} a_{11} & a_{12} & a_{13} & \cdots & a_{n1} & b_1 \\ 0 & a_{22}^1 & a_{23}^1 & \cdots & a_{2n}^1 & b_2^1 \\ \hline 0 & 0 & a_{33}^2 & \cdots & a_{3n}^2 & b_3^2 \\ \hline \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & a_{n3}^2 & \cdots & a_{nn}^2 & b_n^2 \end{array} \right] \end{array} \xrightarrow[3\text{-й шаг}]{\dots} \xrightarrow[(n-1)\text{-й шаг}]{\dots}$$
$$\begin{array}{cccccc} x_1 & x_2 & x_3 & \cdots & x_n & b \\ \hline a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ 0 & a_{22}^1 & a_{23}^1 & \cdots & a_{2n}^1 & b_2^1 \\ 0 & 0 & a_{33}^2 & \cdots & a_{3n}^2 & b_3^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn}^{n-1} & b_n^{n-1} \end{array}$$

В обратном ходе алгоритма Гаусса из последнего уравнения сразу определяется  $x_n$ , из предпоследнего -  $x_{n-1}$  и т.д. Из первого уравнения определяется  $x_1$ .

$$\left\{ \begin{array}{l} a_{nn}^{n-1} x_n = b_n^{n-1} \Rightarrow x_n \\ a_{n-1n-1}^{n-2} x_{n-1} + a_{n-1n}^{n-2} x_n = b_{n-1}^{n-2} \Rightarrow x_{n-1} \\ \dots\dots\dots \\ a_{11} x_1 + \dots + a_{1n} x_n = b_1 \Rightarrow x_1 \end{array} \right.$$

*Замечание 2.* Если элементы какой-либо строки матрицы системы и правая часть в результате преобразований стали равными нулю, то СЛАУ совместна, но имеет бесконечное множество решений, получающееся с помощью метода Гаусса для СЛАУ порядка  $r$ , где  $r$  - ранг матрицы исходной СЛАУ.

$$\det A = (-1)^p a_{11} a_{22}^1 a_{33}^2 \cdot \dots \cdot a_{nn}^{n-1}$$

При этом с помощью множителя  $(-1)^p$ , где  $p$  - число перестановок строк в процессе прямого хода, учитываются соответствующие перемены знаков вследствие перестановок строк.

*Замечание 4.* Метод Гаусса можно применить для обращения невырожденной ( $\det A \neq 0$ ) матрицы.

Действительно, пусть требуется обратить невырожденную матрицу  $A = [a_{ij}]$ ,  $i, j = \overline{1, n}$ . Тогда, сделав обозначение  $A^{-1} = X$ ,  $X = [x_{ij}]$ ,  $i, j = \overline{1, n}$ , можно выписать

матричное уравнение  $AX = E$ , где  $E$  - единичная матрица  $E = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}$ , на основе

которого можно записать цепочку СЛАУ

$$A \cdot \begin{pmatrix} x_{11} \\ x_{21} \\ \dots \\ x_{n1} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \end{pmatrix}, \quad A \cdot \begin{pmatrix} x_{12} \\ x_{22} \\ \dots \\ x_{n2} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ \dots \\ 0 \end{pmatrix}, \quad \dots \quad A \cdot \begin{pmatrix} x_{1n} \\ x_{2n} \\ \dots \\ x_{nn} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 1 \end{pmatrix},$$

каждую из которых можно решить методом Гаусса. При этом, поскольку верхняя треугольная матрица для всех этих СЛАУ будет одной и той же, то метод Гаусса применяется один раз. Строится следующая расширенная матрица:

$$\left( \begin{array}{cccc|cccc} x_{1n} & x_{2n} & \dots & x_{nn} & & & & \\ \dots & \dots & \dots & \dots & & & & \\ x_{12} & x_{22} & \dots & x_{n2} & & & & \\ x_{11} & x_{21} & \dots & x_{n1} & b^1 & b^2 & \dots & b^n \end{array} \right) \xrightarrow{\dots} \left( \begin{array}{cccc|cccc} a_{11} & a_{12} & \dots & a_{1n} & 1 & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & a_{2n} & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} & 0 & 0 & \dots & 1 \end{array} \right) \xrightarrow{\dots}$$

В результате применения  $(n-1)$ -го шага метода Гаусса получаем:

$$\left( \begin{array}{cccc|cccc} x_{1n} & x_{2n} & \dots & x_{nn} & & & & \\ \dots & \dots & \dots & \dots & & & & \\ x_{12} & x_{22} & \dots & x_{n2} & & & & \\ x_{11} & x_{21} & \dots & x_{n1} & b^1 & b^2 & \dots & b^n \end{array} \right) \xrightarrow{\dots} \left( \begin{array}{cccc|cccc} a_{11} & a_{12} & \dots & a_{1n} & b_{11} & b_{12} & \dots & b_{1n} \\ & a_{22}^1 & \dots & a_{2n}^1 & b_{21}^1 & b_{22}^1 & \dots & b_{2n}^1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{nn}^{n-1} & b_{n1}^{n-1} & b_{n2}^{n-1} & \dots & b_{nn}^{n-1} \end{array} \right) \xrightarrow{\dots}$$

При этом первый столбец  $(x_{11} \ x_{21} \ \dots \ x_{n1})^T$  обратной матрицы определяется в обратном ходе метода Гаусса с правой частью  $b^1$ , столбец  $(x_{12} \ x_{22} \ \dots \ x_{n2})^T$  - с правой частью  $b^2$  и так далее. Столбец  $(x_{1n} \ x_{2n} \ \dots \ x_{nn})^T$  определяется с правой частью  $b^n$ .

**Пример 1.1.** Методом Гаусса решить СЛАУ.

$$\begin{cases} 10x_1 + x_2 + x_3 = 12 \\ 2x_1 + 10x_2 + x_3 = 13 \\ 2x_1 + 2x_2 + 10x_3 = 14 \end{cases}$$

Р е ш е н и е.

Прямой ход:

$$\begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & b \end{array} \\ \left( \begin{array}{ccc|c} \mathbf{10} & 1 & 1 & 12 \\ 2 & 10 & 1 & 13 \\ 2 & 2 & 10 & 14 \end{array} \right) \xrightarrow[1\text{-й шаг}]{(-2/10); (-2/10)} \end{array}$$

$$\begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & b \end{array} \\ \left( \begin{array}{ccc|c} 10 & 1 & 1 & 12 \\ 0 & \mathbf{9,8} & 0,8 & 10,6 \\ 0 & 1,8 & 9,8 & 11,6 \end{array} \right) \xrightarrow[2\text{-й шаг}]{(-1,8/9,8)} \begin{array}{cccc} x_1 & x_2 & x_3 & b \\ \left( \begin{array}{ccc|c} 10 & 1 & 1 & 12 \\ 0 & 9,8 & 0,8 & 10,6 \\ 0 & 0 & 9,653 & 9,653 \end{array} \right) \end{array} \end{array}$$

Обратный ход:

$$\begin{aligned} 9,653x_3 &= 9,653, & x_3 &= 1 \\ 9,8x_2 + 0,8x_3 &= 10,6, & x_2 &= 1 \\ 10x_1 + x_2 + x_3 &= 12, & x_1 &= 1. \end{aligned}$$

Ответ:  $x_1 = x_2 = x_3 = 1$ .

**Пример 1.2.** Методом Гаусса вычислить определитель матрицы и обратить матрицу СЛАУ из примера 1.1.

Р е ш е н и е.

$$A = \begin{pmatrix} 10 & 1 & 1 \\ 2 & 10 & 1 \\ 2 & 2 & 10 \end{pmatrix}; \quad \det A = 10 \cdot 9.8 \cdot 9.65 = 945.994 \quad (\text{точное значение } 946).$$

Прямой ход.

$$\begin{array}{c} \begin{array}{cccc} x_{13} & x_{23} & x_{33} & \end{array} \\ \begin{array}{ccc} x_{12} & x_{22} & x_{23} \end{array} \\ \begin{array}{cccc} x_{11} & x_{21} & x_{31} & b^1 \ b^2 \ b^3 \end{array} \\ \left( \begin{array}{ccc|ccc} \mathbf{10} & 1 & 1 & 1 & 0 & 0 \\ 2 & 10 & 1 & 0 & 1 & 0 \\ 2 & 2 & 10 & 0 & 0 & 1 \end{array} \right) \xrightarrow[1\text{-й шаг}]{(-2/10); (-2/10)} \end{array}$$

$$\begin{array}{ccc}
x_{13} & x_{23} & x_{33} \\
x_{12} & x_{22} & x_{23} \\
x_{11} & x_{21} & x_{31} & b^1 & b^2 & b^3
\end{array}
\left( \begin{array}{ccc|ccc}
10 & 1 & 1 & 1 & 0 & 0 \\
0 & \mathbf{9,8} & 0,8 & -0,2 & 1 & 0 \\
0 & 1,8 & 9,8 & -0,2 & 0 & 1
\end{array} \right) (-1,8/9,8) \xrightarrow[2-й \text{ шаг}]{2-й}$$

$$\begin{array}{ccc}
x_{13} & x_{23} & x_{33} \\
x_{12} & x_{22} & x_{32} \\
x_{11} & x_{21} & x_{31} & b^1 & b^2 & b^3
\end{array}
\left( \begin{array}{ccc|ccc}
10 & 1 & 1 & 1 & 0 & 0 \\
0 & 9,8 & 0,8 & -0,2 & 1 & 0 \\
0 & 0 & 9,653 & -0,163 & -0,184 & 1
\end{array} \right)$$

Обратный ход:

$$\begin{cases} 9,653x_{31} = -0,163 \\ 9,8x_{21} + 0,8x_{31} = -0,2 \\ 10x_{11} + x_{21} + x_{31} = 1 \end{cases}
\begin{cases} 9,653x_{32} = -0,184 \\ 9,8x_{22} + 0,8x_{32} = 1 \\ 10x_{12} + x_{22} + x_{32} = 0 \end{cases}
\begin{cases} 9,653x_{33} = 1 \\ 9,8x_{23} + 0,8x_{33} = 0 \\ 10x_{13} + x_{23} + x_{33} = 0 \end{cases}$$

$$\text{Отсюда } A^{-1} = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix} = \begin{pmatrix} 0,104 & -0,0085 & -0,0095 \\ -0,019 & 0,104 & -0,0085 \\ -0,0169 & -0,019 & 0,104 \end{pmatrix}$$

$$\text{Проверка: } A \cdot A^{-1} = \begin{pmatrix} 10 & 1 & 1 \\ 2 & 10 & 1 \\ 2 & 2 & 10 \end{pmatrix} \cdot \begin{pmatrix} 0,104 & -0,0085 & -0,0095 \\ -0,019 & 0,104 & -0,0085 \\ -0,0169 & -0,019 & 0,104 \end{pmatrix} = \begin{pmatrix} 1,004 & 0 & 0,0005 \\ 0,001 & 1,004 & 0 \\ 0,001 & 0,001 & 1,004 \end{pmatrix},$$

т.е. с точностью до ошибок округления получена единичная матрица.

*Замечание 5.* Компьютерная реализация метода Гаусса часто осуществляется с использованием **LU-разложения матриц**.

$LU$  – разложение матрицы  $A$  представляет собой разложение матрицы  $A$  в произведение нижней и верхней треугольных матриц, т.е.

$$A = LU,$$

где  $L$  – нижняя треугольная матрица (матрица, у которой все элементы, находящиеся выше главной диагонали равны нулю,  $l_{ij} = 0$  при  $i < j$ ),  $U$  – верхняя треугольная матрица (матрица, у которой все элементы, находящиеся ниже главной диагонали равны нулю,  $u_{ij} = 0$  при  $i > j$ ).

$LU$  – разложение может быть построено с использованием описанного выше метода Гаусса. Рассмотрим  $k$ -ый шаг метода Гаусса, на котором осуществляется обнуление поддиагональных элементов  $k$ -го столбца матрицы  $A^{(k-1)}$ . Как было описано выше, с этой целью используется следующая операция:

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - \mu_i^{(k)} a_{kj}^{(k-1)}, \quad \mu_i^{(k)} = \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}, \quad i = \overline{k+1, n}, \quad j = \overline{k, n}.$$

В терминах матричных операций такая операция эквивалентна умножению  $A^{(k)} = M_k A^{(k-1)}$ , где элементы матрицы  $M_k$  определяются следующим образом

$$m_{ij}^k = \begin{cases} 1, & i = j \\ 0, & i \neq j, \quad j \neq k \\ -\mu_{k+1}^{(k)}, & i \neq j, \quad j = k \end{cases}.$$

Т.е. матрица  $M_k$  имеет вид

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -\mu_{k+1}^{(k)} & 1 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & -\mu_n^{(k)} & 0 & 0 & 1 \end{pmatrix}.$$

При этом выражение для обратной операции запишется в виде  $A^{(k-1)} = M_k^{-1} A^{(k)}$ , где

$$M_k^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \mu_{k+1}^{(k)} & 1 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \mu_n^{(k)} & 0 & 0 & 1 \end{pmatrix}$$

В результате прямого хода метода Гаусса получим  $A^{(n-1)} = U$ ,

$$A = A^{(0)} = M_1^{-1} A^{(1)} = M_1^{-1} M_2^{-1} A^{(2)} = M_1^{-1} M_2^{-1} \dots M_{n-1}^{-1} A^{(n-1)},$$

где  $A^{(n-1)} = U$  - верхняя треугольная матрица, а  $L = M_1^{-1} M_2^{-1} \dots M_{n-1}^{-1}$  - нижняя треугольная

матрица, имеющая вид  $L =$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ \mu_2^{(1)} & 1 & 0 & 0 & 0 & 0 \\ \mu_3^{(1)} & \mu_3^{(2)} & 1 & 0 & 0 & 0 \\ \dots & \dots & \dots & \mu_{k+1}^{(k)} & 1 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \mu_n^{(1)} & \mu_n^{(2)} & \mu_n^{(k)} & \mu_n^{(k+1)} & \dots & \mu_n^{(n-1)} & 1 \end{pmatrix}.$$

Таким образом, искомое разложение  $A = LU$  получено.

В частности, для рассмотренного выше примера 1.1.  $LU$  – разложение матрицы  $A$

имеет вид  $A = \begin{pmatrix} 10 & 1 & 1 \\ 2 & 10 & 1 \\ 2 & 2 & 10 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0,2 & 1 & 0 \\ 0,2 & 0,18 & 1 \end{pmatrix} \cdot \begin{pmatrix} 10 & 1 & 1 \\ 0 & 9,8 & 0,8 \\ 0 & 0 & 9,65 \end{pmatrix} = LU$

В дальнейшем  $LU$  – разложение может быть эффективно использовано при решении систем линейных алгебраических уравнений вида  $Ax = b$ . Действительно, подставляя  $LU$  – разложение в СЛАУ, получим  $LUx = b$ , или  $Ux = L^{-1}b$ . Т.е. процесс решения СЛАУ сводится к двум простым этапам.

На первом этапе решается СЛАУ  $Lz = b$ . Поскольку матрица системы - нижняя треугольная, решение можно записать в явном виде:

$$z_1 = b_1, \quad z_i = b_i - \sum_{j=1}^{i-1} l_{ij} z_j, \quad i = \overline{2, n}.$$

На втором этапе решается СЛАУ  $Ux = z$  с верхней треугольной матрицей. Здесь, как и на предыдущем этапе, решение представляется в явном виде:

$$x_n = \frac{z_n}{u_{nn}}, \quad x_i = \frac{1}{u_{ii}} \left( z_i - \sum_{j=i+1}^n u_{ij} x_j \right), \quad i = \overline{n-1, 1}.$$

Отметим, что второй этап эквивалентен обратному ходу методу Гаусса, тогда как первый соответствует преобразованию правой части СЛАУ в процессе прямого хода.



### 1.1.2. Метод прогонки

Метод прогонки является одним из эффективных методов решения СЛАУ с трех - диагональными матрицами, возникающих при конечно-разностной аппроксимации задач для обыкновенных дифференциальных уравнений (ОДУ) и уравнений в частных производных второго порядка и является частным случаем метода Гаусса. Рассмотрим следующую СЛАУ:

$$(1.1) \quad a_1 = 0 \left\{ \begin{array}{l} b_1 x_1 + c_1 x_2 = d_1 \\ a_2 x_1 + b_2 x_2 + c_2 x_3 = d_2 \\ \quad \quad \quad a_3 x_2 + b_3 x_3 + c_3 x_4 = d_3 \\ \dots\dots\dots \\ \quad \quad \quad a_{n-1} x_{n-2} + b_{n-1} x_{n-1} + c_{n-1} x_n = d_{n-1} \\ \quad \quad \quad a_n x_{n-1} + b_n x_n = d_n, \quad c_n = 0, \end{array} \right.$$

решение которой будем искать в виде

$$(1.2) \quad x_i = P_i x_{i+1} + Q_i, \quad i = \overline{1, n}$$

где  $P_i, Q_i, i = \overline{1, n}$  - прогоночные коэффициенты, подлежащие определению. Для их определения выразим из первого уравнения СЛАУ (1.1)  $x_1$  через  $x_2$ , получим:

$$(1.3) \quad x_1 = \frac{-c_1}{b_1} x_2 + \frac{d_1}{b_1} = P_1 x_2 + Q_1,$$

откуда

$$P_1 = \frac{-c_1}{b_1}, \quad Q_1 = \frac{d_1}{b_1}.$$

Из второго уравнения СЛАУ (1.1) с помощью (1.3) выразим  $x_2$  через  $x_3$ , получим:

$$x_2 = \frac{-c_2}{b_2 + a_2 P_1} x_3 + \frac{d_2 - a_2 Q_1}{b_2 + a_2 P_1} = P_2 x_3 + Q_2,$$

откуда

$$P_2 = \frac{-c_2}{b_2 + a_2 P_1}, \quad Q_2 = \frac{d_2 - a_2 Q_1}{b_2 + a_2 P_1}.$$

Продолжая этот процесс, получим из  $i$ -го уравнения СЛАУ (1.1):

$$x_i = \frac{-c_i}{b_i + a_i P_{i-1}} x_{i+1} + \frac{d_i - a_i Q_{i-1}}{b_i + a_i P_{i-1}},$$

следовательно

$$P_i = \frac{-c_i}{b_i + a_i P_{i-1}}, \quad Q_i = \frac{d_i - a_i Q_{i-1}}{b_i + a_i P_{i-1}}.$$

Из последнего уравнения СЛАУ имеем

$$x_n = \frac{-c_n}{b_n + a_n P_{n-1}} x_{n+1} + \frac{d_n - a_n Q_{n-1}}{b_n + a_n P_{n-1}} = 0 \cdot x_{n+1} + Q_n,$$

ТО ЕСТЬ

$$P_n = 0 \text{ (т.к. } c_n = 0), Q_n = \frac{d_n - a_n Q_{n-1}}{b_n + a_n P_{n-1}} = x_n.$$

Таким образом, прямой ход метода прогонки по определению прогоночных коэффициентов  $P_i, Q_i, i = \overline{1, n}$  завершен. В результате прогоночные коэффициенты вычисляются по следующим формулам:

$$P_i = \frac{-c_i}{b_i + a_i P_{i-1}}, \quad Q_i = \frac{d_i - a_i Q_{i-1}}{b_i + a_i P_{i-1}}, \quad i = \overline{2, n-1}; \quad (1.4)$$

$$P_1 = \frac{-c_1}{b_1}, \quad Q_1 = \frac{d_1}{b_1}, \text{ так как } a_1 = 0, \quad i = 1; \quad (1.5)$$

$$P_n = 0, \text{ т.к. } c_n = 0, Q_n = \frac{d_n - a_n Q_{n-1}}{b_n + a_n P_{n-1}}, \quad i = n. \quad (1.6)$$

Обратный ход метода прогонки осуществляется в соответствии с выражением (1.2)

$$(1.7) \quad \begin{cases} x_n = P_n x_{n+1} + Q_n = 0 \cdot x_{n+1} + Q_n = Q_n \\ x_{n-1} = P_{n-1} x_n + Q_{n-1} \\ x_{n-1} = P_{n-2} x_{n-1} + Q_{n-2} \\ \dots\dots\dots \\ x_1 = P_1 x_2 + Q_1. \end{cases}$$

Формулы (1.4)-(1.7) - формулы *правой прогонки*.

Аналогично, начиная с последнего уравнения СЛАУ (1.1) можно вывести формулы *левой прогонки*.

Общее число операций в методе прогонки равно  $8n+1$ , т.е. пропорционально числу уравнений. Такие методы решения СЛАУ называют *экономичными*. Для сравнения число операций в методе Гаусса пропорционально  $n^3$  [1].

Для устойчивости метода прогонки (1.4)-(1.7) достаточно выполнение следующих условий:

$$a_i \neq 0, \quad c_i \neq 0, \quad i = \overline{2, n-1}$$

$$|b_i| \geq |a_i| + |c_i|, \quad i = \overline{1, n},$$

(1.8)

причем строгое неравенство имеет место хотя бы при одном  $i$ . Здесь устойчивость понимается в смысле ненакопления погрешности решения в ходе вычислительного процесса при малых погрешностях входных данных (правых частей и элементов матрицы СЛАУ).

**Пример 1.3.** Методом прогонки решить СЛАУ

$$\begin{cases} 8x_1 - 2x_2 = 6 \\ -x_1 + 6x_2 - 2x_3 = 3 \\ 2x_2 + 10x_3 - 4x_4 = 8 \\ -x_3 + 6x_4 = 5 \end{cases}$$

**Решение.**

$$P_1 = \frac{-c_1}{b_1} = \frac{2}{8} = 0,25, \quad Q_1 = \frac{d_1}{b_1} = 0,75;$$

$$P_2 = \frac{-c_2}{b_2 + a_2 P_1} = \frac{2}{6 - 1 \cdot 0,25} = 0,3478, \quad Q_2 = \frac{d_2 - a_2 Q_1}{b_2 + a_2 P_1} = \frac{(3 + 1 \cdot 0,75)}{5,75} = 0,6522;$$

$$P_3 = \frac{-c_3}{b_3 + a_3 P_2} = 0,374, \quad Q_3 = \frac{d_3 - a_3 Q_2}{b_3 + a_3 P_2} = 0,626;$$

$$P_4 = 0 \quad (c_4 = 0), \quad Q_4 = \frac{d_4 - a_4 Q_3}{b_4 + a_4 P_3} = 1,0;$$

$$x_4 = P_4 x_5 + Q_4 = 1,0, \quad x_3 = P_3 x_4 + Q_3 = 1,0, \quad x_2 = P_2 x_3 + Q_2 = 1,0,$$

$$x_1 = P_1 x_2 + Q_1 = 1,0.$$

### 1.1.3. Нормы векторов и матриц

Для исследования сходимости численных методов решения задач линейной алгебры вводятся понятия нормы векторов и матриц.

**Нормой вектора**  $x = (x_1, x_2, \dots, x_n)^T$  (обозначают  $\|x\|$ ) в  $n$ -мерном вещественном пространстве векторов  $x \in R^n$  называют неотрицательное число, вычисляемое с помощью компонент вектора и обладающее следующими свойствами:

- а)  $\|x\| \geq 0$  ( $\|x\| = 0$  тогда и только тогда, когда  $x$  - нулевой вектор  $x = \mathcal{O}$ );
- б)  $\|\alpha \cdot x\| = |\alpha| \cdot \|x\|$  для любых действительных чисел  $\alpha$ ;
- в)  $\|x + y\| \leq \|x\| + \|y\|$ .

**Нормой матрицы**  $A_{n \times n}$  (обозначается  $\|A\|$ ) с вещественными элементами в пространстве матриц называют неотрицательное число, вычисляемое с помощью элементов матрицы и обладающее следующими свойствами:

- а)  $\|A\| > 0$  ( $\|A\| = 0$  тогда и только тогда, когда  $A$  - нулевая матрица  $A = \Theta$ );
- б)  $\|\alpha \cdot A\| = |\alpha| \cdot \|A\|$  для любых действительных чисел  $\alpha$ ;
- в)  $\|A + B\| \leq \|A\| + \|B\|$  для всех  $n \times n$  матриц  $A$  и  $B$  рассматриваемого пространства;
- г)  $\|A \cdot B\| \leq \|A\| \cdot \|B\|$  для всех  $n \times n$  матриц  $A$  и соответствующих матриц  $B$ .

Как видно из последнего свойства (если в качестве матрицы  $B$  использовать вектор  $x$ ), норма матриц должна быть согласована с нормой векторов. Это согласование осуществляется связью

$$\|Ax\| \leq \|A\| \|x\|. \quad (1.9)$$

Наиболее употребительными являются следующие нормы векторов:

$$\|x\|_1 = \sum_{i=1}^n |x_i|, \quad (1.10)$$

$$\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2} = \sqrt{(x, x)}. \quad (1.11)$$

$$\|x\|_c = \max_i |x_i|, \quad (1.12)$$

Наиболее распространенными согласованными с ними с помощью связи (1.9) нормами матриц будут соответственно:

$$\|A\|_1 = \max_j \sum_{i=1}^n |a_{ij}|, \quad (1.13)$$

$$\|A\|_2 = \sqrt{\sum_{i,j=1}^n a_{ij}^2}. \quad (1.14)$$

$$\|A\|_c = \max_i \sum_{j=1}^n |a_{ij}|, \quad (1.15)$$

Отметим, что норма (1.15) согласована со всеми приведенными выше нормами векторов.

Для исследования погрешностей, возникающих при решении СЛАУ, вводят понятие *числа обусловленности матрицы*  $\text{cond}(A)$ :

$$\text{cond}(A) = \|A\| \cdot \|A^{-1}\|$$

Число обусловленности характеризует степень зависимости относительной погрешности решения СЛАУ от погрешности входных данных (правые части, элементы матрицы). Можно показать что для ненулевых векторов  $x$  справедливы следующие неравенства:

$$\frac{\|\Delta x\|}{\|x\|} \leq \text{cond} A \frac{\|\Delta b\|}{\|b\|}, \quad \frac{\|\Delta x\|}{\|x\|} \leq \text{cond} A \frac{\|\Delta A\|}{\|A + \Delta A\|}$$

Таким образом, чем больше число обусловленности, тем сильнее влияние погрешности входных данных на конечный результат. Матрица считается плохо обусловленной, если  $\text{cond}(A) \gg 1$ .

Если в качестве нормы матрицы принять ее спектральный радиус  $\max_i |\lambda_i|$  (см. раздел 1.2 настоящего пособия), то

$$\text{cond}(A) = \max_i |\lambda_i| \frac{1}{\min_i |\lambda_i|} \geq 1$$

поскольку спектральный радиус обратной матрицы  $A^{-1}$  равен обратной величине минимального собственного значения исходной матрицы.

#### **Пример 1.4.**

Для матрицы  $A$  и вектора  $b$  вычислить различные нормы  $\|\cdot\|_1, \|\cdot\|_2, \|\cdot\|_c$ . Проверить выполнение условия согласованности норм  $\|Ax\| \leq \|A\| \|x\|$  для различных комбинаций норм. Вычислить число обусловленности матрицы  $A$ .

$$A = \begin{pmatrix} -1 & 2 \\ 3 & -5 \end{pmatrix}, b = \begin{pmatrix} 3 \\ -4 \end{pmatrix}.$$

### Решение.

Вычислим соответствующие нормы:

$$\|b\|_1 = |3| + |-4| = 7, \|b\|_2 = (3^2 + (-4)^2)^{1/2} = 5, \|b\|_c = \max(|3|, |-4|) = 4.$$

$$\|A\|_1 = \max(|-1| + |3|, |2| + |-5|) = 7, \|A\|_2 = ((-1)^2 + 3^2 + 2^2 + (-5)^2)^{1/2} = \sqrt{39},$$

$$\|A\|_c = \max(|-1| + |2|, |3| + |-5|) = 8.$$

Для проверки условия согласованности вычислим различные нормы вектора

$$c = Ab = \begin{pmatrix} -11 \\ 29 \end{pmatrix}.$$

$$\|c\|_1 = |-11| + |29| = 40, \|c\|_2 = ((-11)^2 + 29^2)^{1/2} = \sqrt{962}, \|c\|_c = \max(|-11|, |29|) = 29.$$

Легко убедиться в том, что условие согласованности выполняется для согласованных норм:

$$\|c\|_1 = 40 \leq \|A\|_1 \|b\|_1 = 7 \cdot 7 = 49, \|c\|_2 = \sqrt{962} \leq \|A\|_2 \|b\|_2 = \sqrt{39} \cdot 5 = \sqrt{975},$$

$$\|c\|_c = 29 \leq \|A\|_c \|b\|_c = 8 \cdot 4 = 32.$$

Кроме того, известно что матричная норма  $\|A\|_c$  согласована со всеми введенными выше нормами векторов. В данном примере это подтверждается выполнением неравенств:

$$\|c\|_1 = 40 \leq \|A\|_c \|b\|_1 = 8 \cdot 7 = 56, \|c\|_2 = \sqrt{962} \leq \|A\|_c \|b\|_2 = 8 \cdot 5 = 40.$$

В то же время использование ряда других комбинаций норм матрицы и вектора приводит в данном случае к нарушению условия согласованности:

$$\|c\|_c = 29 > \|A\|_1 \|b\|_c = 7 \cdot 4 = 28, \|c\|_c = 29 > \|A\|_2 \|b\|_c = \sqrt{39} \cdot 4.$$

Рассмотренный пример наглядно иллюстрирует важность использования согласованных норм матрицы и вектора.

Вычислим число обусловленности матрицы  $A$ , взяв в качестве нормы матрицы  $\|\cdot\|_c$ . Для этого найдем сначала обратную матрицу:

$$A^{-1} = \begin{pmatrix} 5 & 2 \\ 3 & 1 \end{pmatrix}$$

и вычислим ее норму:

$$\|A^{-1}\|_c = \max(|5| + |2|, |3| + |1|) = 7.$$

В результате

$$\text{cond}(A) = \|A\|_c \|A^{-1}\|_c = 8 \cdot 7 = 56.$$

## 1.1.4. Итерационные методы решения СЛАУ.

### Метод простых итераций

При большом числе уравнений прямые методы решения СЛАУ (за исключением метода прогонки) становятся труднореализуемыми на ЭВМ прежде всего из-за сложности хранения и обработки матриц большой размерности. В то же время характерной особенностью ряда часто встречающихся в прикладных задачах СЛАУ является разреженность матриц. Число ненулевых элементов таких матриц мало по сравнению с

При таком способе приведения исходной СЛАУ к эквивалентному виду метод простых итераций носит название метода Якоби.

В качестве нулевого приближения  $x^{(0)}$  вектора неизвестных примем вектор правых частей  $x^{(0)} = \beta$  или  $(x_1^{(0)} \ x_2^{(0)} \ \dots \ x_n^{(0)})^T = (\beta_1 \ \beta_2 \ \dots \ \beta_n)^T$ . Тогда *метод простых итераций* примет вид:

$$\begin{cases} x^{(0)} = \beta \\ x^{(1)} = \beta + \alpha x^{(0)} \\ x^{(2)} = \beta + \alpha x^{(1)} \\ \dots \\ x^{(k)} = \beta + \alpha x^{(k-1)}. \end{cases} \quad (1.19)$$

Из (1.19) видно преимущество итерационных методов по сравнению, например, с рассмотренным выше методом Гаусса. В вычислительном процессе участвуют только произведения матрицы на вектор, что позволяет работать только с ненулевыми элементами матрицы, значительно упрощая процесс хранения и обработки матриц.

Имеет место следующее достаточное условие сходимости метода простых итераций.

*Метод простых итераций (1.19) сходится к единственному решению СЛАУ (1.17) (а следовательно и к решению исходной СЛАУ (1.16)) при любом начальном приближении  $x^{(0)}$ , если какая-либо норма матрицы  $\alpha$  эквивалентной системы меньше единицы  $\|\alpha\| < 1$ .*

Если используется метод Якоби (выражения (1.18) для эквивалентной СЛАУ), то достаточным условием сходимости является *диагональное преобладание матрицы  $A$* , т.е.

$|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}| \quad \forall i$  (для каждой строки матрицы  $A$  модули элементов, стоящих на главной диагонали, больше суммы модулей недиагональных элементов). Очевидно, что в этом случае  $\|\alpha\|_c$  меньше единицы и, следовательно, итерационный процесс (1.19) сходится.

Приведем также необходимое и достаточное условие сходимости метода простых итераций. *Для сходимости итерационного процесса (1.19) необходимо и достаточно, чтобы спектр матрицы  $\alpha$  эквивалентной системы лежал внутри круга с радиусом, равным единице.*

При выполнении достаточного условия сходимости оценка погрешности решения на  $k$ -ой итерации дается выражением:

$$\|x^{(k)} - x^*\| \leq \varepsilon^{(k)} = \frac{\|\alpha\|}{1 - \|\alpha\|} \|x^{(k)} - x^{(k-1)}\|, \quad (1.20)$$

где  $x^*$  - точное решение СЛАУ.



Процесс итераций останавливается при выполнении условия  $\varepsilon^{(k)} \leq \varepsilon$ , где  $\varepsilon$  - задаваемая вычислителем точность.

Принимая во внимание, что из (1.20) следует неравенство

$$\|x^{(k)} - x^*\| \leq \frac{\|\alpha\|^k}{1 - \|\alpha\|} \|x^{(1)} - x^{(0)}\|,$$

можно получить априорную оценку необходимого для достижения заданной точности числа итераций. При использовании в качестве начального приближения вектора  $\beta$  такая оценка определится неравенством:

$$\frac{\|\alpha\|^{k+1}}{1 - \|\alpha\|} \|\beta\| \leq \varepsilon,$$

откуда получаем априорную оценку числа итераций  $k$  при  $\|\alpha\| < 1$

$$k + 1 \geq \frac{\lg \varepsilon - \lg \|\beta\| + \lg(1 - \|\alpha\|)}{\lg \|\alpha\|}.$$

Следует подчеркнуть, что это неравенство дает завышенное число итераций  $k$ , поэтому редко используется на практике.

Замечание. Поскольку  $\|\alpha\| < 1$  является только достаточным (не необходимым) условием сходимости метода простых итераций, то итерационный процесс может сходиться и в случае, если оно не выполнено. Тогда критерием окончания итераций может служить неравенство  $\|x^{(k)} - x^{(k-1)}\| \leq \varepsilon$ .

**Пример 1.5.** Методом простых итераций с точностью  $\varepsilon = 0,01$  решить СЛАУ.

$$\begin{cases} 10x_1 + x_2 + x_3 = 12 \\ 2x_1 + 10x_2 + x_3 = 13 \\ 2x_1 + 2x_2 + 10x_3 = 14 \end{cases}$$

**Р е ш е н и е.**

Приведем СЛАУ к эквивалентному виду:

$$\begin{cases} x_1 = 1,2 - 0,1x_2 - 0,1x_3 \\ x_2 = 1,3 - 0,2x_1 - 0,1x_3 \\ x_3 = 1,4 - 0,2x_1 - 0,2x_2 \end{cases}$$

или  $x = \beta + \alpha x$

$$\text{где } \alpha = \begin{pmatrix} 0 & -0,1 & -0,1 \\ -0,2 & 0 & -0,1 \\ -0,2 & -0,2 & 0 \end{pmatrix}; \quad \beta = (1,2 \quad 1,3 \quad 1,4)^T;$$

$\|\alpha\|_c = 0,4 < 1$ , следовательно достаточное условие сходимости метода простых итераций выполнено.

Итерационный процесс выглядит следующим образом.

$$x^{(0)} = \beta; \quad x^{(1)} = \beta + \alpha\beta = (0,93 \quad 0,92 \quad 0,9)^T; \quad \varepsilon^{(1)} = 0.333 > \varepsilon;$$

$$x^{(2)} = \beta + \alpha x^{(1)} = (1,018 \quad 1,024 \quad 1,03)^T; \quad \varepsilon^{(2)} = 0.0867 > \varepsilon$$

$$x^{(3)} = \beta + \alpha x^{(2)} = (0,9946 \quad 0,9934 \quad 0,9916)^T; \quad \varepsilon^{(3)} = 0.0256 > \varepsilon$$

$$x^{(4)} = \beta + \alpha x^{(3)} = (1,0015 \quad 1,00192 \quad 1,0024)^T; \quad \varepsilon^{(4)} = 0.0072 < \varepsilon.$$

Таким образом, вычислительный процесс завершен за 4 итерации. Отметим, что точное решение исходной СЛАУ в данном случае известно  $x^* = (1 \quad 1 \quad 1)^T$ . Отсюда следует, что заданной точности  $\varepsilon = 0,01$  удовлетворяло решение, полученное уже на третьей итерации. Но в силу использования для вычисления погрешности оценочного выражения (1.20) (видно, что в данном случае  $\|x^{(3)} - x^*\| \leq \varepsilon^{(3)}$ , при этом  $\varepsilon^{(3)} > \varepsilon$ , хотя  $\|x^{(3)} - x^*\| \leq \varepsilon$ ) процесс останавливается только на четвертой итерации.

Отметим также, что априорная оценка необходимого количества итераций в данной задаче дает:  $k+1 \geq (-2 + \lg 0,6 - \lg 1,4) / \lg 0,4 = 5,95$ , т.е. для достижения точности  $\varepsilon = 0,01$ , согласно априорной оценке, необходимо сделать не менее пяти итераций, что иллюстрирует характерную для априорной оценки тенденцию к завышению числа итераций.

## Метод Зейделя решения СЛАУ

Метод простых итераций довольно медленно сходится. Для его ускорения существует метод Зейделя, заключающийся в том, что при вычислении компонента  $x_i^{k+1}$  вектора неизвестных на  $(k+1)$ -й итерации используются  $x_1^{k+1}, x_2^{k+1}, \dots, x_{i-1}^{k+1}$ , уже вычисленные на  $(k+1)$ -й итерации. Значения остальных компонент  $x_{i+1}^k, x_{i+2}^k, \dots, x_n^k$  берутся из предыдущей итерации. Так же как и в методе простых итераций строится эквивалентная СЛАУ (1.17) и за начальное приближение принимается вектор правых частей  $x^0 = (\beta_1 \quad \beta_2 \quad \dots \quad \beta_n)^T$ . Тогда метод Зейделя для известного вектора  $(x_1^k \quad x_2^k \quad \dots \quad x_n^k)^T$  на  $k$ -ой итерации имеет вид:

$$\begin{cases} x_1^{k+1} = \beta_1 + \alpha_{11}x_1^k + \alpha_{12}x_2^k + \dots + \alpha_{1n}x_n^k \\ x_2^{k+1} = \beta_2 + \alpha_{21}x_1^{k+1} + \alpha_{22}x_2^k + \dots + \alpha_{2n}x_n^k \\ x_3^{k+1} = \beta_3 + \alpha_{31}x_1^{k+1} + \alpha_{32}x_2^{k+1} + \alpha_{33}x_3^k + \dots + \alpha_{3n}x_n^k \\ \dots \\ x_n^{k+1} = \beta_n + \alpha_{n1}x_1^{k+1} + \alpha_{n2}x_2^{k+1} + \dots + \alpha_{nn-1}x_{n-1}^{k+1} + \alpha_{nn}x_n^k. \end{cases}$$

Из этой системы видно, что  $x^{k+1} = \beta + Bx^{k+1} + Cx^k$ , где  $B$  - нижняя треугольная матрица с диагональными элементами, равными нулю, а  $C$  - верхняя треугольная матрица с диагональными элементами, отличными от нуля,  $\alpha = B + C$ . Следовательно

$$(E - B)x^{k+1} = Cx^k + \beta,$$

откуда

$$x^{k+1} = (E - B)^{-1} Cx^k + (E - B)^{-1} \beta.$$

Таким образом, метод Зейделя является методом простых итераций с матрицей правых частей  $\alpha = (E - B)^{-1} C$  и вектором правых частей  $(E - B)^{-1} \beta$  и, следовательно, сходимость и погрешность метода Зейделя можно исследовать с помощью формул, выведенных для метода простых итераций, в которых вместо матрицы  $\alpha$  подставлена матрица  $(E - B)^{-1} C$ , а вместо вектора правых частей – вектор  $(E - B)^{-1} \beta$ . Для практических вычислений важно, что в качестве достаточных условий сходимости метода Зейделя могут быть использованы условия, приведенные выше для метода простых итераций ( $\|\alpha\| < 1$  или, если используется эквивалентная СЛАУ в форме (1.18) – диагональное преобладание матрицы  $A$ ). В случае выполнения этих условий для оценки погрешности на  $k$ -ой итерации можно использовать выражение

$$\varepsilon^{(k)} = \frac{\|C\|}{1 - \|\alpha\|} \|x^{(k)} - x^{(k-1)}\|.$$

Отметим, что как и метод простых итераций, метод Зейделя может сходиться и при нарушении условия  $\|\alpha\| < 1$ . В этом случае  $\varepsilon^{(k)} = \|x^{(k)} - x^{(k-1)}\|$ .

**Пример 1.6.** Методом Зейделя решить СЛАУ из примера 1.5.

**Р е ш е н и е.**

Приведение СЛАУ к эквивалентному виду аналогично примеру (1.5). Диагональное преобладание элементов исходной матрицы СЛАУ гарантирует сходимость метода Зейделя.

Итерационный процесс выглядит следующим образом:

$$x^{(0)} = (1,2 \quad 1,3 \quad 1,4)^T$$

$$\begin{cases} x_1^{(1)} = 1,2 - 0,1 \cdot 1,3 - 0,1 \cdot 1,4 = 0,93 \\ x_2^{(1)} = 1,3 - 0,2 \cdot 0,93 - 0,1 \cdot 1,4 = 0,974 \\ x_3^{(1)} = 1,4 - 0,2 \cdot 0,93 - 0,2 \cdot 0,974 = 1,0192 \end{cases}$$

$$\begin{cases} x_1^{(2)} = 1,2 - 0,1 \cdot 0,974 - 0,1 \cdot 1,0192 = 1,0007 \\ x_2^{(2)} = 1,3 - 0,2 \cdot 1,0007 - 0,1 \cdot 1,0192 = 0,998 \\ x_3^{(2)} = 1,4 - 0,2 \cdot 1,0007 - 0,2 \cdot 0,998 = 1,0003 \end{cases}$$

Таким образом, уже на второй итерации погрешность  $\|x^{(2)} - x^*\| < 10^{-2} = \varepsilon$ , т.е. метод Зейделя в данном случае сходится быстрее метода простых итераций.

## 1.2. Численные методы решения задач на собственные значения и собственные векторы матриц

### 1.2.1. Основные определения и спектральные свойства матриц

Рассмотрим матрицу  $A_{n \times n}$  в  $n$ -мерном вещественном пространстве  $R^n$  векторов

$$x = (x_1 \ x_2 \ \dots \ x_n)^T.$$

1. Собственным вектором  $x$  матрицы  $A$  называется ненулевой вектор ( $x \neq 0$ ), удовлетворяющий равенству

$$Ax = \lambda x, \quad (1.21)$$

где  $\lambda$  - собственное значение матрицы  $A$ , соответствующее рассматриваемому собственному вектору.

2. Собственные значения матрицы  $A$  с действительными элементами могут быть вещественными различными, вещественными кратными, комплексными попарно сопряженными, комплексными кратными.

3. Классический способ нахождения собственных значений и собственных векторов известен и заключается в следующем:

для однородной СЛАУ, полученной из (1.21)

$$(A - \lambda E)x = 0, \quad 0 = (0 \ 0 \ \dots \ 0)^T$$

ненулевые решения ( $x \neq 0$ , а именно такие решения и находятся) имеют место при

$$\det(A - \lambda E) = 0, \quad (1.22)$$

причем уравнение (1.22) называют *характеристическим уравнением*, а выражение в левой части - *характеристическим многочленом*;

каким-либо способом находят решения  $\lambda_1, \lambda_2, \dots, \lambda_n$  алгебраического уравнения (1.22)  $n$ -й степени (предположим, что они вещественны и различны);

решая однородную СЛАУ (1.22) для различных собственных значений  $\lambda_j$ ,  $j = \overline{1, n}$ ,

$$(A - \lambda_j E)x^j = 0, \quad j = \overline{1, n},$$

получаем линейно независимые собственные векторы  $x^j$ ,  $j = \overline{1, n}$ , соответствующие собственным значениям  $\lambda_j$ ,  $j = \overline{1, n}$ .

4. Попарно различным собственным значениям соответствуют линейно независимые собственные векторы;  $k$ -кратному корню характеристического уравнения (1.22), построенного для произвольной матрицы  $A_{n \times n}$ , соответствует не более  $k$  ( $\leq k$ ) линейно независимых собственных векторов. Если количество линейно независимых собственных векторов матрицы  $A_{n \times n}$  совпадает с размерностью пространства  $R^n$ , то их можно принять за новый базис, в котором матрица  $A_{n \times n}$  примет диагональный вид

$$\Lambda = U^{-1} \cdot A \cdot U, \quad (1.23)$$

на главной диагонали которой находятся собственные значения, а столбцы матрицы преобразования  $U$  являются собственными векторами матрицы  $A$  (матрицы  $\Lambda$  и  $A$ , удовлетворяющие равенству (1.23), называются *подобными*). Собственные значения *подобных* матриц  $\Lambda$  и  $A$  совпадают.

5. Симметрическая матрица  $A$  ( $A = A^T$ ) имеет полный спектр  $\lambda_j$ ,  $j = \overline{1, n}$  вещественных собственных значений; положительно определенная симметрическая матрица ( $A = A^T$ ,  $(Ax, x) > 0$ ) имеет полный спектр вещественных положительных собственных значений;  $k$ -кратному корню характеристического уравнения (1.22) симметрической матрицы соответствует ровно  $k$  линейно независимых собственных векторов; симметрическая матрица имеет ровно  $n$  ортогональных собственных векторов, приняв которые за новый базис (т.е. построив матрицу преобразования  $U$ , взяв в качестве ее столбцов координатные столбцы собственных векторов), можно преобразовать симметрическую матрицу  $A$  к диагональному виду с помощью преобразования (1.23); для симметрической матрицы  $A$  матрица преобразования  $U$  в (1.23) является ортогональной  $U^{-1} = U^T$  и, следовательно, преобразование (1.23) имеет вид

$$\Lambda = U^T \cdot A \cdot U. \quad (1.24)$$

### 1.2.2. Метод вращений Якоби численного решения задач на собственные значения и собственные векторы матриц

Метод вращений Якоби применим только для симметрических матриц  $A_{n \times n}$  ( $A = A^T$ ) и решает полную проблему собственных значений и собственных векторов таких матриц. Он основан на отыскании с помощью итерационных процедур матрицы  $U$  в преобразовании подобия  $\Lambda = U^{-1}AU$ , а поскольку для симметрических матриц  $A$  матрица преобразования подобия  $U$  является ортогональной ( $U^{-1} = U^T$ ), то  $\Lambda = U^T AU$ , где  $\Lambda$  - диагональная матрица с собственными значениями на главной диагонали

$$\Lambda = \begin{pmatrix} \lambda_1 & \dots & 0 \\ \dots & \ddots & \dots \\ 0 & \dots & \lambda_n \end{pmatrix}.$$

Пусть дана симметрическая матрица  $A$ . Требуется для нее вычислить с точностью  $\varepsilon$  все собственные значения и соответствующие им собственные векторы. Алгоритм метода вращения следующий:

Пусть известна матрица  $A^{(k)}$  на  $k$ -й итерации, при этом для  $k=0$   $A^{(0)} = A$ .

1. Выбирается максимальный по модулю недиагональный элемент  $a_{ij}^{(k)}$  матрицы  $A^{(k)}$  ( $|a_{ij}^{(k)}| = \max_{l < m} |a_{lm}^{(k)}|$ ).

2. Ставится задача найти такую ортогональную матрицу  $U^{(k)}$ , чтобы в результате преобразования подобия  $A^{(k+1)} = U^{(k)T} A^{(k)} U^{(k)}$  произошло обнуление элемента  $a_{ij}^{(k+1)}$  матрицы  $A^{(k+1)}$ . В качестве ортогональной матрицы выбирается матрица вращения, имеющая следующий вид:

$$U^k = \begin{pmatrix} & i & j & & \\ & \vdots & \vdots & & \\ & \ddots & \vdots & & 0 \\ & 1 & \vdots & & \\ \dots & \cos \varphi^{(k)} & \dots & -\sin \varphi^{(k)} & \dots & i \\ & \vdots & 1 & \vdots & & \\ & \vdots & \ddots & \vdots & & \\ & \vdots & & 1 & \vdots & \\ \dots & \sin \varphi^{(k)} & \dots & \cos \varphi^{(k)} & \dots & j \\ & \vdots & & \vdots & 1 & \\ 0 & \vdots & & \vdots & \ddots & \\ & \vdots & & \vdots & & 1 \end{pmatrix},$$

В матрице вращения на пересечении  $i$ -й строки и  $j$ -го столбца находится элемент  $u_{ij}^{(k)} = -\sin \varphi^{(k)}$ , где  $\varphi^{(k)}$  - угол вращения, подлежащий определению. Симметрично относительно главной диагонали ( $j$ -я строка,  $i$ -й столбец) расположен элемент  $u_{ji}^{(k)} = \sin \varphi^{(k)}$ ; Диагональные элементы  $u_{ii}^{(k)}$  и  $u_{jj}^{(k)}$  равны соответственно  $u_{ii}^{(k)} = \cos \varphi^{(k)}$ ,  $u_{jj}^{(k)} = \cos \varphi^{(k)}$ ; другие диагональные элементы  $u_{mm}^{(k)} = 1, m = \overline{1, n}, m \neq i, m \neq j$ ; остальные элементы в матрице вращения  $U^{(k)}$  равны нулю.

Угол вращения  $\varphi^{(k)}$  определяется из условия  $a_{ij}^{(k+1)} = 0$ :

$$\varphi^{(k)} = \frac{1}{2} \operatorname{arctg} \frac{2a_{ij}^{(k)}}{a_{ii}^{(k)} - a_{jj}^{(k)}},$$

причем если  $a_{ii}^{(k)} = a_{jj}^{(k)}$ , то  $\varphi^{(k)} = \frac{\pi}{4}$ .

3. Строится матрица  $A^{(k+1)}$

$$A^{(k+1)} = U^{(k)T} A^{(k)} U^{(k)},$$

в которой элемент  $a_{ij}^{(k+1)} \approx 0$ .

В качестве критерия окончания итерационного процесса используется условие малости суммы квадратов внедиагональных элементов:

$$t(A^{(k+1)}) = \left( \sum_{l,m; l < m} (a_{lm}^{(k+1)})^2 \right)^{1/2}.$$

Если  $t(A^{(k+1)}) > \varepsilon$ , то итерационный процесс

$$A^{(k+1)} = U^{(k)T} A^{(k)} U^{(k)} = U^{(k)T} U^{(k-1)T} \dots U^{(0)T} A^{(0)} U^{(0)} U^{(1)} \dots U^{(k)}$$

продолжается. Если  $t(A^{(k+1)}) < \varepsilon$ , то итерационный процесс останавливается, и в качестве искомых собственных значений принимаются  $\lambda_1 \approx a_{11}^{(k+1)}, \lambda_2 \approx a_{22}^{(k+1)}, \dots, \lambda_n \approx a_{nn}^{(k+1)}$ .

Координатными столбцами собственных векторов матрицы  $A$  в единичном базисе будут столбцы матрицы  $U = U^{(0)} U^{(1)} \dots U^{(k)}$ , т.е.

$$(x^1)^T = (u_{11} \ u_{21} \ \dots \ u_{n1}), \quad (x^2)^T = (u_{12} \ u_{22} \ \dots \ u_{n2}), \quad (x^n)^T = (u_{1n} \ u_{2n} \ \dots \ u_{nn}),$$

причем эти собственные векторы будут ортогональны между собой, т.е.  $(x^l, x^m) \approx 0, l \neq m$ .

**Пример 1.7.** С точностью  $\varepsilon = 0,3$  вычислить собственные значения и собственные векторы матрицы

$$A = \begin{bmatrix} 4 & 2 & 1 \\ 2 & 5 & 3 \\ 1 & 3 & 6 \end{bmatrix} \equiv A^{(0)}.$$

**Решение.**

1). Выбираем максимальный по модулю внедиагональный элемент матрицы  $A^{(0)}$ , т.е. находим  $a_{ij}^{(0)}$ , такой что  $|a_{ij}^{(0)}| = \max_{l < m} |a_{lm}^{(0)}|$ . Им является элемент  $a_{23}^{(0)} = 3$ .

2). Находим соответствующую этому элементу матрицу вращения:

$$U^{(0)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi^{(0)} & -\sin \varphi^{(0)} \\ 0 & \sin \varphi^{(0)} & \cos \varphi^{(0)} \end{bmatrix}; \varphi^{(0)} = \frac{1}{2} \operatorname{arctg} \frac{2 \cdot 3}{5-6} =$$

$$= -0,7033; \sin \varphi^{(0)} = -0,65; \cos \varphi^{(0)} = 0,76;$$

$$U^{(0)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0,76 & 0,65 \\ 0 & -0,65 & 0,76 \end{bmatrix}; .$$

3). Вычисляем матрицу  $A^{(1)}$ :

$$A^{(1)} = U^{(0)T} A^{(0)} U = \begin{bmatrix} 4 & 0,87 & 2,06 \\ 0,87 & 2,46 & -0,03 \\ 2,06 & -0,03 & 8,54 \end{bmatrix}.$$

В полученной матрице с точностью до ошибок округления элемент  $a_{23}^{(1)} = 0$ .

$$t(A^{(1)}) = \left( \sum_{l,m;l < m} (a_{lm}^{(1)})^2 \right)^{1/2} = (0,87^2 + 2,06^2 + (-0,03)^2)^{1/2} > \varepsilon, \text{ следовательно}$$

итерационный процесс необходимо продолжить.

Переходим к следующей итерации ( $k = 1$ ):

$$a_{13}^{(1)} = 2,06; \quad \left( |a_{13}^{(1)}| = \max_{l,m;l < m} |a_{lm}^{(1)}| \right).$$

$$U^{(1)} = \begin{bmatrix} \cos \varphi^{(1)} & 0 & -\sin \varphi^{(1)} \\ 0 & 1 & 0 \\ \sin \varphi^{(1)} & 0 & \cos \varphi^{(1)} \end{bmatrix};$$

$$\varphi^{(1)} = \frac{1}{2} \operatorname{arctg} \frac{2 \cdot 2,06}{4-8,54} = -0,3693; \quad \sin \varphi^{(1)} = -0,361; \quad \cos \varphi^{(1)} = 0,933;$$

$$U^{(1)} = \begin{bmatrix} 0,933 & 0 & 0,361 \\ 0 & 1 & 0 \\ -0,361 & 0 & 0,933 \end{bmatrix}; \quad A^{(2)} = U^{(1)T} A^{(1)} U^{(1)} = \begin{bmatrix} 3,19 & 0,819 & 0,005 \\ 0,819 & 2,46 & 0,28 \\ 0,005 & 0,28 & 9,38 \end{bmatrix}.$$

$$t(A^{(2)}) = \left( \sum_{l,m;l < m} (a_{lm}^{(2)})^2 \right)^{1/2} = (0,819^2 + 0,28^2 + 0,005^2)^{1/2} > \varepsilon.$$

Переходим к следующей итерации ( $k = 2$ )

$$a_{12}^{(2)} = 0,819; \quad \left( |a_{12}^{(2)}| = \max_{l,m;l < m} |a_{lm}^{(2)}| \right)$$

$$\varphi^{(2)} = \frac{1}{2} \operatorname{arctg} \frac{2 \cdot 0,819}{3,19-2,46} = 0,5758; \quad \sin \varphi^{(2)} = 0,5445; \quad \cos \varphi^{(2)} = 0,8388.$$



$$U^{(2)} = \begin{bmatrix} 0,8388 & -0,5445 & 0 \\ 0,5445 & 0,8388 & 0 \\ 0 & 0 & 1 \end{bmatrix}; A^{(3)} = U^{(2)T} A^{(2)} U^{(2)} = \begin{bmatrix} 3,706 & 0,0003 & 0,1565 \\ 0,0003 & 1,929 & 0,232 \\ 0,1565 & 0,232 & 9,38 \end{bmatrix};$$

$$t(A^{(3)}) = (0,0003^2 + 0,1565^2 + 0,232^2)^{1/2} = 0,07839^{1/2} < \varepsilon.$$

Таким образом в качестве искоемых собственных значений могут быть приняты диагональные элементы матрицы  $A^{(3)}$ :

$$\lambda_1 \approx 3,706; \lambda_2 \approx 1,929; \lambda_3 \approx 9,38.$$

Собственные векторы определяются из произведения

$$U^{(0)}U^{(1)}U^{(2)} = \begin{bmatrix} 0,78 & -0,5064 & 0,361 \\ 0,2209 & 0,7625 & 0,6 \\ -0,58 & -0,398 & 0,7 \end{bmatrix}; x^1 = \begin{bmatrix} 0,78 \\ 0,2209 \\ -0,58 \end{bmatrix}; x^2 = \begin{bmatrix} -0,5064 \\ 0,7625 \\ -0,398 \end{bmatrix}; x^3 = \begin{bmatrix} 0,361 \\ 0,6 \\ 0,7 \end{bmatrix}.$$

Полученные собственные векторы ортогональны в пределах заданной точности, т.е.  $(x^1, x^2) = -0,00384$ ;  $(x^1, x^3) = 0,0081$ ;  $(x^2, x^3) = -0,0039$ .

### 1.2.3. Частичная проблема собственных значений и собственных векторов матрицы. Степенной метод

Рассмотренный метод вращения решает полную проблему собственных значений и собственных векторов матриц (симметрических) в том смысле, что определяются все собственные значения и собственные векторы.

Зачастую не нужно находить все собственные значения (спектр) и все собственные векторы, а необходимо найти максимальное и минимальное из них. Существует степенной метод по определению спектрального радиуса матрицы, т.е. максимального собственного значения матрицы, и соответствующего ему собственного вектора.

Пусть дана матрица  $A$  и пусть ее собственные значения упорядочены по абсолютным величинам:

$$|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n| \quad (1.25)$$

Тогда, выбрав некоторый вектор  $y^{(0)}$ , например, вектор, компоненты которого равны единице  $y^{(0)} = (1 \ 1 \dots 1)^T$ , можно для определения  $\lambda_1$  построить следующий итерационный процесс:

$$\begin{aligned} y^{(1)} &= Ay^{(0)}, \quad \lambda_1^{(1)} = \frac{y_j^{(1)}}{y_j^{(0)}} \\ y^{(2)} &= Ay^{(1)}, \quad \lambda_1^{(2)} = \frac{y_j^{(2)}}{y_j^{(1)}}; \\ &\dots\dots\dots \\ y^{(k)} &= Ay^{(k-1)}, \quad \lambda_1^{(k)} = \frac{y_j^{(k)}}{y_j^{(k-1)}}; \end{aligned} \quad (1.26)$$

где  $y_j^{(k-1)}$ ,  $y_j^{(k)}$  - соответствующие компоненты векторов  $y^{(k-1)}$ ,  $y^{(k)}$ . При этом в качестве номера  $j$  может использоваться любое число из диапазона  $j = \overline{1, n}$ .

В связи с тем, что вектор  $y^{(k)}$  на  $k$ -ой итерации может быть представлен в виде  $y^{(k)} = Ay^{(k-1)} = A^k y^{(0)}$ , рассматриваемый итерационный процесс носит название "степенной метод". При выполнении условий ( 1.25 ) итерационный процесс сходится к искомому собственному значению  $\lambda_1$  и соответствующему собственному вектору, причем скорость сходимости определяется отношением  $\frac{|\lambda_2|}{|\lambda_1|}$  (чем оно меньше, тем выше скорость сходимости).

В качестве критерия завершения вычислений используется следующее условие:

$$\varepsilon^{(k)} = |\lambda_1^{(k)} - \lambda_1^{(k-1)}| \leq \varepsilon, \text{ где } \varepsilon - \text{ задаваемая вычислителем точность расчета.}$$

**Пример 1.8.**

Вычислить спектральный радиус матрицы  $A = \begin{pmatrix} 5 & 1 & 2 \\ 1 & 4 & 1 \\ 2 & 1 & 3 \end{pmatrix}$  с точностью  $\varepsilon = 0,1$ .

В качестве начального приближения собственного вектора возьмем  $y^{(0)} = (111)^T$ . Реализуем итерационный процесс (1.26), полагая  $j = 1$ .

$$y^{(1)} = Ay^{(0)} = \begin{pmatrix} 8 & 6 & 6 \end{pmatrix}^T, \quad \lambda_1^{(1)} = \frac{y_1^{(1)}}{y_1^{(0)}} = \frac{8}{1} = 8;$$

$$y^{(2)} = Ay^{(1)} = \begin{pmatrix} 58 & 38 & 40 \end{pmatrix}^T, \quad \lambda_1^{(2)} = \frac{y_1^{(2)}}{y_1^{(1)}} = \frac{58}{8} = 7,25;$$

$$\varepsilon^{(2)} = |\lambda_1^{(2)} - \lambda_1^{(1)}| = 0,75 > \varepsilon;$$

$$y^{(3)} = Ay^{(2)} = \begin{pmatrix} 480 & 250 & 274 \end{pmatrix}^T, \quad \lambda_1^{(3)} = \frac{y_1^{(3)}}{y_1^{(2)}} = \frac{480}{58} = 7,034;$$

$$\varepsilon^{(3)} = |\lambda_1^{(3)} - \lambda_1^{(2)}| = 0,216 > \varepsilon;$$

$$y^{(4)} = Ay^{(3)} = \begin{pmatrix} 2838 & 1682 & 1888 \end{pmatrix}^T, \quad \lambda_1^{(4)} = \frac{y_1^{(4)}}{y_1^{(3)}} = \frac{2838}{408} = 6,9559;$$

$$\varepsilon^{(4)} = |\lambda_1^{(4)} - \lambda_1^{(3)}| = 0,078 < \varepsilon.$$

Таким образом, полученное на 4-ой итерации значение  $\lambda_1^{(4)} = 6,9559$  удовлетворяет заданной точности и может быть взято в качестве приближенного значения  $\lambda_1$ . Искомое значение спектрального радиуса  $\rho(A) = \max_i |\lambda_i| = |\lambda_1| = 6,9559$ .

Рассмотренный выше пример наглядно иллюстрирует существенный недостаток алгоритма (1.26), связанный с сильным возрастанием компонентов итерированного вектора

$y^{(k)}$  в ходе итерационного процесса. Видно, что  $\left| \frac{y_j^{(k)}}{y_j^{(k-1)}} \right| \approx |\lambda_1|$ . Во избежание

неограниченного возрастания (при  $|\lambda_1| > 1$ ) или убывания (при  $|\lambda_1| < 1$ ) компонентов  $y^{(k)}$  по мере увеличения числа итераций  $k$  обычно при проведении компьютерных расчетов применяется степенной метод с нормировкой итерированного вектора. С этой целью алгоритм (1.26) модифицируется следующим образом:

$$z^{(k)} = Ay^{(k-1)}, \quad \lambda_1^{(k)} = \frac{z_j^{(k)}}{y_j^{(k-1)}}, \quad y^{(k)} = \frac{z^{(k)}}{\|z^{(k)}\|} \quad (1.27)$$

При этом в качестве начального приближения  $y^{(0)}$  берется вектор с единичной нормой.

Широко распространена также версия степенного метода, использующая скалярные произведения:

$$z^{(k)} = Ay^{(k-1)}, \quad y^{(k)} = \frac{z^{(k)}}{\|z^{(k)}\|}, \quad \lambda_1^{(k)} = (y^{(k)}, Ay^{(k)}) \quad (1.28)$$

#### 1.2.4. QR-алгоритм нахождения собственных значений матриц

При решении полной проблемы собственных значений для несимметричных матриц эффективным является подход, основанный на приведении матриц к подобным, имеющим треугольный или квазитреугольный вид. Одним из наиболее распространенных методов этого класса является QR-алгоритм, позволяющий находить как вещественные, так и комплексные собственные значения.

В основе QR-алгоритма лежит представление матрицы в виде  $A = QR$ , где  $Q$  - ортогональная матрица ( $Q^{-1} = Q^T$ ), а  $R$  - верхняя треугольная. Такое разложение существует для любой квадратной матрицы. Одним из возможных подходов к построению QR разложения является использование преобразования Хаусхолдера, позволяющего обратить в нуль группу поддиагональных элементов столбца матрицы.

Преобразование Хаусхолдера осуществляется с использованием матрицы Хаусхолдера, имеющей следующий вид:

$$H = E - \frac{2}{v^T v} v v^T, \quad (1.29)$$

где  $v$  - произвольный ненулевой вектор-столбец,  $E$  - единичная матрица,  $v v^T$  - квадратная матрица того же размера.

Легко убедиться, что любая матрица такого вида является симметричной и ортогональной. При этом произвол в выборе вектора  $v$  дает возможность построить матрицу, отвечающую некоторым дополнительным требованиям.

Рассмотрим случай, когда необходимо обратить в нуль все элементы какого-либо вектора кроме первого, т.е. построить матрицу Хаусхолдера такую, что

$$\tilde{b} = Hb, \quad b = (b_1, b_2, \dots, b_n)^T, \quad \tilde{b} = (\tilde{b}_1, 0, \dots, 0)^T.$$

Тогда вектор  $v$  определится следующим образом:

$$v = b + \text{sign}(b_1) \|b\|_2 e_1, \quad (1.30)$$

где  $\|b\|_2 = \left( \sum_i b_i^2 \right)^{1/2}$  - евклидова норма вектора,  $e_1 = (1, 0, \dots, 0)^T$ .

Применяя описанную процедуру с целью обнуления поддиагональных элементов каждого из столбцов исходной матрицы, можно за фиксированное число шагов получить ее QR – разложение.

Рассмотрим подробнее реализацию данного процесса.

Положим  $A_0 = A$  и построим преобразование Хаусхолдера  $H_1$  ( $A_1 = H_1 A_0$ ), переводящее матрицу  $A_0$  в матрицу  $A_1$  с нулевыми элементами первого столбца под главной диагональю:

$$A_0 = \begin{pmatrix} a_{11}^0 & a_{12}^0 & \dots & a_{1n}^0 \\ a_{21}^0 & a_{22}^0 & \dots & a_{2n}^0 \\ \dots & \dots & \dots & \dots \\ a_{n1}^0 & a_{n2}^0 & \dots & a_{nn}^0 \end{pmatrix} \xrightarrow{H_1} A_1 = \begin{pmatrix} a_{11}^1 & a_{12}^1 & \dots & a_{1n}^1 \\ 0 & a_{22}^1 & \dots & a_{2n}^1 \\ \dots & \dots & \dots & \dots \\ 0 & a_{n2}^1 & \dots & a_{nn}^1 \end{pmatrix}$$

Ясно, что матрица Хаусхолдера  $H_1$  должна определяться по первому столбцу матрицы  $A_0$ , т.е. в качестве вектора  $b$  в выражении (1.30) берется вектор  $(a_{11}^0, a_{21}^0, \dots, a_{n1}^0)^T$ . Тогда компоненты вектора  $v$  вычисляются следующим образом:

$$v_1^1 = a_{11}^0 + \text{sign}(a_{11}^0) \left( \sum_{j=1}^n (a_{j1}^0)^2 \right)^{1/2},$$

$$v_i^1 = a_{i1}^0, \quad i = \overline{2, n}.$$

Матрица Хаусхолдера  $H_1$  вычисляется согласно (1.29):

$$H_1 = E - 2 \frac{v^1 v^{1T}}{v^{1T} v^1}.$$

На следующем, втором, шаге рассматриваемого процесса строится преобразование Хаусхолдера  $H_2$  ( $A_2 = H_2 A_1$ ), обнуляющее расположенные ниже главной диагонали элементы второго столбца матрицы  $A_1$ . Взяв в качестве вектора  $b$  вектор  $(a_{22}^1, a_{32}^1, \dots, a_{n2}^1)^T$  размерности  $n-1$ , получим следующие выражения для компонентов вектора  $v$ :

$$v_1^2 = 0,$$

$$v_2^2 = a_{22}^1 + \text{sign}(a_{22}^1) \left( \sum_{j=2}^n (a_{j2}^1)^2 \right)^{1/2},$$

$$v_i^2 = a_{i2}^1, \quad i = \overline{3, n}.$$

Повторяя процесс  $n-1$  раз, получим искомое разложение  $A = QR$ , где

$$Q = (H_{n-1} H_{n-2} \dots H_0)^T = H_1 H_2 \dots H_{n-1}, \quad R = A_{n-1}.$$

Следует отметить определенное сходство рассматриваемого процесса с алгоритмом Гаусса. Отличие заключается в том, что здесь обнуление поддиагональных элементов соответствующего столбца осуществляется с использованием ортогонального преобразования.

Процедура  $QR$  - разложения многократно используется в  $QR$  -алгоритме вычисления собственных значений. Строится следующий итерационный процесс:

$$A^{(0)} = A,$$

$$A^{(0)} = Q^{(0)} R^{(0)} - \text{производится } QR - \text{разложение},$$

$$A^{(1)} = R^{(0)} Q^{(0)} - \text{производится перемножение матриц},$$

.....

$$A^{(k)} = Q^{(k)} R^{(k)} - \text{разложение},$$

$$A^{(k+1)} = R^{(k)} Q^{(k)} \text{ перемножение.}$$

Таким образом, каждая итерация реализуется в два этапа. На первом этапе осуществляется разложение матрицы  $A^{(k)}$  в произведение ортогональной  $Q^{(k)}$  и верхней треугольной  $R^{(k)}$  матриц, а на втором – полученные матрицы перемножаются в обратном порядке.

Нетрудно показать подобие матриц  $A^{(k+1)}$  и  $A^{(k)}$ . Действительно, учитывая ортогональность  $Q^{(k)}$  ( $Q^{(k)T} Q^{(k)} = E$ ), можно записать:

$$A^{(k+1)} = R^{(k)} Q^{(k)} = Q^{(k)T} Q^{(k)} R^{(k)} Q^{(k)} = Q^{(k)T} A^{(k)} Q^{(k)}.$$

Аналогично можно показать, что любая из матриц  $A^{(k)}$  ортогонально подобна матрице  $A$ .

При отсутствии у матрицы кратных собственных значений последовательность  $A^{(k)}$  сходится к верхней треугольной матрице (в случае, когда все собственные значения вещественны) или к верхней квазитреугольной матрице (если имеются комплексно-сопряженные пары собственных значений).

Таким образом, каждому вещественному собственному значению будет соответствовать столбец со стремящимися к нулю поддиагональными элементами и в качестве критерия сходимости итерационного процесса для таких собственных значений

можно использовать следующее неравенство:  $\left( \sum_{l=m+1}^n (a_{lm}^{(k)})^2 \right)^{1/2} \leq \varepsilon$ . При этом

соответствующее собственное значение принимается равным диагональному элементу данного столбца.

Каждой комплексно-сопряженной паре соответствует диагональный блок размерностью  $2 \times 2$ , т.е. матрица  $A^{(k)}$  имеет блочно-диагональную структуру. Принципиально то, что элементы этих блоков изменяются от итерации к итерации без видимой закономерности, в то время как комплексно-сопряженные собственные значения,

определяемые каждым блоком, имеют тенденцию к сходимости. Это обстоятельство необходимо учитывать при формировании критерия выхода из итерационного процесса. Если в ходе итераций прослеживается комплексно-сопряженная пара собственных значений, соответствующая блоку, образуемому элементами  $j$ -го и  $(j+1)$ -го столбцов  $a_{jj}^{(k)}, a_{jj+1}^{(k)}, a_{j+1j}^{(k)}, a_{j+1j+1}^{(k)}$ , то, несмотря на значительное изменение в ходе итераций самих этих элементов, собственные значения, соответствующие данному блоку и определяемые из решения квадратного уравнения  $(a_{jj}^{(k)} - \lambda^{(k)})(a_{j+1j+1}^{(k)} - \lambda^{(k)}) = a_{jj+1}^{(k)} a_{j+1j}^{(k)}$ , начиная с некоторого  $k$ , отличаются незначительно. В качестве критерия окончания итераций для таких блоков может быть использовано следующее условие  $|\lambda^{(k)} - \lambda^{(k-1)}| \leq \varepsilon$ .

*Замечание.* Существенным недостатком рассмотренного выше алгоритма является большое число операций (пропорционально  $n^3$ , где  $n$  - размерность матрицы), необходимое для  $QR$  - факторизации матрицы на каждой итерации. Эффективность  $QR$  - алгоритма может быть повышена, если предварительно с помощью преобразования подобия привести матрицу к верхней Хессенберговой форме, в которой равны нулю все элементы, находящиеся ниже главной диагонали за исключением элементов первой поддиагонали. Иными словами предварительно производится следующая операция:

$$A^{(0)} = H^T A H,$$

где  $A^{(0)}$  - матрица Хессенберга, имеющая следующую структуру (знак  $x$  обозначает ненулевые элементы):

$$\begin{pmatrix} x & x & x & \dots & x & x \\ x & x & x & \dots & x & x \\ 0 & x & x & \dots & x & x \\ 0 & 0 & x & \dots & x & x \\ \dots & \dots & \dots & \dots & x & x \\ 0 & 0 & 0 & 0 & x & x \end{pmatrix},$$

Здесь принципиально то, что в дальнейшем, в ходе  $QR$  - итераций, матрицы  $A^{(k)}$  сохраняют верхнюю Хессенбергову форму, что позволяет более экономно проводить их  $QR$  - разложение. Подробное изложение данного вопроса можно найти, например, в [2].

**Пример 1.9.**

Используя преобразование Хаусхолдера, построить  $QR$  - разложение матрицы

$$A = \begin{pmatrix} 1 & 3 & 1 \\ 1 & 1 & 4 \\ 4 & 3 & 1 \end{pmatrix}.$$

**Решение.**

1. Положим  $A_0 = A$  и найдем ортогональную матрицу Хаусхолдера  $H_1$ , такую что в матрице  $A_1 = H_1 A_0$  все поддиагональные элементы первого столбца равны нулю. С этой целью компоненты вектора  $v$  определим, используя элементы первого столбца матрицы  $A_0$ :

$$v_1^1 = a_{11}^0 + \text{sign}(a_{11}^0) \left( \sum_{j=1}^3 (a_{j1}^0)^2 \right)^{1/2} = 1 + (1 + 1 + 4^2)^{1/2} = 5,24,$$

$$v_2^1 = a_{21}^0 = 1,$$

$$v_3^1 = a_{31}^0 = 4.$$

В результате получен вектор  $v^1 = (5,24 \ 1 \ 4)^T$ .

Найдем соответствующую этому вектору матрицу Хаусхолдера:

$$H_1 = E - 2 \frac{v^1 v^{1T}}{v^{1T} v^1} = \begin{pmatrix} -0,24 & -0,24 & -0,94 \\ -0,24 & 0,96 & -0,18 \\ -0,94 & -0,18 & 0,28 \end{pmatrix}.$$

В заключение первого шага вычислим матрицу  $A_1$ :

$$A_1 = H_1 A_0 = \begin{pmatrix} -4,24 & -3,77 & -2,12 \\ 0 & -0,29 & 3,40 \\ 0 & -2,17 & -1,38 \end{pmatrix}.$$

Таким образом, после первого шага получена матрица с нулевыми поддиагональными элементами в первом столбце.

2. На втором шаге проделаем аналогичную процедуру, обнуляя поддиагональный элемент второго столбца.

$$v_1^2 = 0,$$

$$v_2^2 = a_{22}^1 + \text{sign}(a_{22}^1) \left( \sum_{j=2}^3 (a_{j2}^1)^2 \right)^{1/2} = -0,29 - (0,29^2 + 2,17^2)^{1/2} = -2,48,$$

$$v_3^2 = a_{32}^1 = -2,17.$$

Т.е. искомый вектор  $v^2 = (0 \ -2,48 \ -2,17)^T$ .

Далее найдем соответствующую ему матрицу Хаусхолдера:

$$H_2 = E - 2 \frac{v^2 v^{2T}}{v^{2T} v^2} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -0,13 & -0,99 \\ 0 & -0,99 & 0,13 \end{pmatrix}$$

и вычислим матрицу  $A_2$ :

$$A_2 = H_2 A_1 = \begin{pmatrix} -4,24 & -3,77 & -2,12 \\ 0 & 2,19 & 0,91 \\ 0 & 0 & -3,56 \end{pmatrix}.$$



Таким образом, исходная матрица  $A$  приведена к верхнему треугольному виду, т.е. получена матрица  $R = A_2$  искомого разложения.

Резльтирующая ортогональная матрица преобразования  $Q$  получается в результате перемножения матриц  $H_i, i = 1, 2$ :

$$Q = H_1 H_2 = \begin{pmatrix} -0,24 & 0,97 & 0,11 \\ -0,24 & 0,05 & -0,97 \\ -0,94 & -0,25 & 0,22 \end{pmatrix}.$$

В заключение выпишем окончательный результат  $A = QR$  в явном виде:

$$A = \begin{pmatrix} 1 & 3 & 1 \\ 1 & 1 & 4 \\ 4 & 3 & 1 \end{pmatrix} = \begin{pmatrix} -0,24 & 0,97 & 0,11 \\ -0,24 & 0,05 & -0,97 \\ -0,94 & -0,25 & 0,22 \end{pmatrix} \times \begin{pmatrix} -4,24 & -3,77 & -2,12 \\ 0 & 2,19 & 0,91 \\ 0 & 0 & -3,56 \end{pmatrix}.$$

### Пример 1.10.

С помощью  $QR$  - алгоритма вычислить собственные значения матрицы  $A$  из предыдущего примера с точностью  $\varepsilon = 0,01$ .

#### Решение.

1. Положим  $A^{(0)} = A$  и найдем  $QR$  - разложение этой матрицы  $A^{(0)} = Q^{(0)} R^{(0)}$ .

Эта процедура подробно рассмотрена в предыдущем примере. Получены следующие  $Q^{(0)}, R^{(0)}$ :

$$Q^{(0)} = \begin{pmatrix} -0,24 & 0,97 & 0,11 \\ -0,24 & 0,05 & -0,97 \\ -0,94 & -0,25 & 0,22 \end{pmatrix}, \quad R^{(0)} = \begin{pmatrix} -4,24 & -3,77 & -2,12 \\ 0 & 2,19 & 0,91 \\ 0 & 0 & -3,56 \end{pmatrix}.$$

Матрицу  $A^{(1)}$  определим перемножением полученных в результате  $QR$  - разложения матриц в обратном порядке  $A^{(1)} = R^{(0)} Q^{(0)}$ :

$$A^{(1)} = \begin{pmatrix} 3,89 & -3,75 & 2,74 \\ -1,38 & -0,12 & -1,92 \\ 3,35 & 0,9 & -0,77 \end{pmatrix}.$$

Первая итерация завершена. Поддиагональные элементы матрицы  $A^{(1)}$  достаточно велики, поэтому итерационный процесс необходимо продолжить.

2. а). Находим  $QR$  - разложение  $A^{(1)} = Q^{(1)} R^{(1)}$  (используя преобразование Хаусхолдера аналогично примеру):

$$Q^{(1)} = \begin{pmatrix} -0,73 & 0,68 & 0,05 \\ 0,26 & 0,21 & 0,94 \\ -0,63 & -0,7 & 0,33 \end{pmatrix}, \quad R^{(1)} = \begin{pmatrix} -5,32 & 2,14 & -2,02 \\ 0 & 3,21 & -2,0 \\ 0 & 0 & -1,93 \end{pmatrix}.$$

б). Перемножая полученные выше матрицы в обратном порядке находим матрицу  $A^{(2)}$ :

$$A^{(2)} = \begin{pmatrix} 5,72 & 1,75 & 1,09 \\ 2,09 & -2,08 & 2,37 \\ 1,22 & -1,36 & -0,64 \end{pmatrix}.$$

Продолжая итерационный процесс получим соответственно на 6-ой и 7-ой итерациях следующие матрицы:

$$A^{(6)} = \begin{pmatrix} 6,34 & 0,94 & -0,73 \\ 0,034 & -2,53 & 1,69 \\ 0,023 & -1,86 & -0,81 \end{pmatrix}, \quad A^{(7)} = \begin{pmatrix} 6,34 & 0,27 & 1,13 \\ -0,0014 & -2,01 & -2,58 \\ 0,0006 & 0,98 & -1,33 \end{pmatrix}.$$

Видно, что поддиагональные элементы первого столбца становятся достаточно малыми, и, следовательно, диагональный элемент  $a_{11}^{(7)}$  может быть принят в качестве собственного значения. В то же время отчетливо прослеживается комплексно-сопряженная пара собственных значений, соответствующая блоку, образуемому элементами второго и третьего столбцов  $a_{22}^{(k)}, a_{23}^{(k)}, a_{32}^{(k)}, a_{33}^{(k)}$ . Несмотря на значительное изменение в ходе итераций самих этих элементов, собственные значения, соответствующие данному блоку и определяемые из решения квадратного уравнения  $(a_{22}^{(k)} - \lambda^{(k)})(a_{33}^{(k)} - \lambda^{(k)}) = a_{23}^{(k)} a_{32}^{(k)}$ , меняются незначительно (в пределах допустимой погрешности). Таким образом,

окончательное решение задачи можно записать в виде:  $\lambda_1 = \lambda_1^{(7)} = 6,34$ ,  
 $\lambda_2 = \lambda_2^7 = -1,67 + 1,55i$ ,  $\lambda_3 = \lambda_3^7 = -1,67 - 1,55i$ .