

На первом шаге алгоритма Гаусса выберем диагональный элемент $a_{11} \neq 0$ (если он равен 0, то первую строку переставляем с какой-либо ниже лежащей 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:

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

↑ *Ведущий столбец*

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 *ниже лежащую* 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 $a_{3j}^1, j = \overline{3, n}, \quad b_3^1, \quad \Sigma_3^1 \quad - \quad \text{элементы} \quad a_{3j}^2 = a_{3j}^1 + a_{2j}^1 \left(-\frac{a_{32}^1}{a_{22}^1} \right), \quad j = \overline{3, n},$

$$b_3^2 = b_3^1 + b_2^1 \left(-\frac{a_{32}^1}{a_{22}^1} \right), \text{. И так далее. Умножая ведущую строку на число } \left(-\frac{a_{n2}^1}{a_{22}^1} \right),$$

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 , Σ_n^1 -

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

строки матрицы неизменными, получим в результате второго шага алгоритма Гаусса следующую матрицу:

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

\uparrow Ведущий столбец

После $(n-1)$ -го шага алгоритма Гаусса получаем следующую расширенную матрицу, содержащую верхнюю треугольную матрицу СЛАУ:

$$\begin{array}{cccccc} x_1 & x_2 & x_3 & \Lambda & x_n & b \\ \hline a_{11} & a_{12} & a_{13} & \Lambda & a_{1n} & b_1 \\ 0 & a_{22}^1 & a_{23}^1 & \Lambda & a_{2n}^1 & b_2^1 \\ 0 & 0 & a_{33}^2 & \Lambda & a_{3n}^2 & b_3^2 \\ \Lambda & \Lambda & \Lambda & \Lambda & \Lambda & \Lambda \\ 0 & 0 & 0 & \Lambda & a_{nn}^{n-1} & b_n^{n-1} \end{array}$$

Прямой ход алгоритма Гаусса завершен.

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 .

[illegible]

Замечание 1. Если элементы какой-либо строки матрицы системы в результате преобразований стали равными нулю, а правая часть не равна нулю, то СЛАУ несовместна, поскольку не выполняются условия теоремы Кронекера-Капелли.

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.

Замечание 3. В результате прямого хода метода Гаусса можно вычислить определитель матрицы A исходной СЛАУ:

$$\det A = (-1)^p a_{11}^1 a_{22}^2 a_{33}^3 \dots a_{nn}^{n-1}$$

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:

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

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

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

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.

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

Решение.

Прямой ход:

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

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

Обратный ход:

$$9,653x_3 = 9,653, \quad x_3 = 1$$

$$9,8x_2 + 0,8x_3 = 10,6, \quad x_2 = 1$$

$$10x_1 + x_2 + x_3 = 12, \quad x_1 = 1.$$

Ответ: $x_1 = x_2 = x_3 = 1$.

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

Решение.

$$A = \begin{pmatrix} 10 & 1 & 1 \\ 2 & 10 & 1 \\ 2 & 2 & 10 \end{pmatrix}; \quad \det A = 10 \cdot 9,8 \cdot 9,65 = 945,994 \quad (\text{точное значение } 946).$$

Прямой ход.

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

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

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

Обратный ход:

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

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

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

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:

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

решение которой будем искать в виде

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

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:

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

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

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

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:

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

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

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

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

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

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 [2]:

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

$$|b_i| \geq |a_i| + |c_i|, \quad i = \overline{1, n}, \quad (1.8)$$

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 $x = 0$);
- б) $\|\alpha \cdot x\| = |\alpha| \cdot \|x\|$ для любых действительных чисел α ;
- 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 $A = 0$);
- б) $\|\alpha \cdot A\| = |\alpha| \cdot \|A\|$ для любых действительных чисел α ;
- 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* $\text{cond}(A)$ [1]:

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

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 .

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

Решение.

Вычислим соответствующие нормы:

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

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

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

Для проверки условия согласованности вычислим различные нормы вектора

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

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

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

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

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

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

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

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

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

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 их размерностью. Для решения СЛАУ с разреженными матрицами предпочтительнее использовать итерационные методы.

Методы последовательных приближений, в которых при вычислении последующего приближения решения используются предыдущие, уже известные приближенные решения, называются *итерационными*.

Рассмотрим СЛАУ

[illegible]

с невырожденной матрицей ($\det A \neq 0$).

Приведем СЛАУ к эквивалентному виду

[illegible]

или в векторно-матричной форме

$$x = \beta + \alpha x, \quad x = \begin{pmatrix} x_1 \\ \mathbf{M} \\ x_n \end{pmatrix}, \quad \beta = \begin{pmatrix} \beta_1 \\ \mathbf{M} \\ \beta_n \end{pmatrix}, \quad \alpha = \begin{pmatrix} \alpha_{11} & \Lambda & \alpha_{1n} \\ \mathbf{M} & \Lambda & \mathbf{M} \\ \alpha_{n1} & \mathbf{K} & \alpha_{nn} \end{pmatrix}.$$

Такое приведение может быть выполнено различными способами. Одним из наиболее распространенных является следующий.

Разрешим систему (1.16) относительно неизвестных при ненулевых диагональных элементах $a_{ii} \neq 0$, $i = \overline{1, n}$ (если какой-либо коэффициент на главной диагонали равен нулю, достаточно соответствующее уравнение поменять местами

с любым другим уравнением). Получим следующие выражения для компонентов вектора β и матрицы α эквивалентной системы:

$$\beta_i = \frac{b_i}{a_{ii}}; \quad \alpha_{ij} = -\frac{a_{ij}}{a_{ii}}, \quad i, j = \overline{1, n}, \quad i \neq j; \quad \alpha_{ij} = 0, \quad i = j, \quad i = \overline{1, n}. \quad (1.18)$$

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 $(x_1^{(0)} \quad x_2^{(0)} \quad \dots \quad x_n^{(0)})^T = (\beta_1 \quad \beta_2 \quad \dots \quad \beta_n)^T$. Тогда *метод простых итераций* примет вид:

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

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 [1].

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 α 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 α 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 ε - 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 β 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.

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

Решение.

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:

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

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 α 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.

Найдите больше информации на сайте **Учитель.ру** (www.uchites.ru)!