

Модификации метода Ньютона

№	Наименование	Формулы
1	Упрощенный метод Ньютона	$x_{n+1} = x_n - f(x_n) / f'(x_0), n = 0,1,2,\dots$
2	Метод хорд	$x_{n+1} = x_n - \frac{c - x_n}{f(c) - f(x_n)} f(x_n), n = 0,1,2,\dots$ <p>c – фиксированная точка из окрестности корня</p>
3	Метод секущих	$x_{n+1} = x_n - \frac{x_{n-1} - x_n}{f(x_{n-1}) - f(x_n)} f(x_n), n = 0,1,2,\dots$
4	Метод Стеффенсена	$x_{n+1} = x_n - \frac{f(x_n)}{f(x_n + f(x_n)) - f(x_n)} f(x_n),$ <p>$n = 0,1,2,\dots$</p>
5	Модифицированный метод Ньютона для поиска кратных корней	$x_{n+1} = x_n - m \frac{f(x_n)}{f'(x_n)}, n = 0,1,2,\dots,$ <p>где m – кратность вычисляемого корня</p>

Задачи вычислительной алгебры. Прямые и итерационные методы. Метод исключения неизвестных (метод Гаусса) решения систем линейных алгебраических уравнений (СЛАУ). Схема единственного деления. Метод Гаусса с выбором главного элемента. LU – разложение матрицы. Методы вращений, квадратного корня.

К вычислительным задачам линейной алгебры относят задачи решения систем линейных алгебраических уравнений (СЛАУ) $A\vec{x} = \vec{b}$, вычисления обратных матриц A^{-1} , вычисления определителей $|A|$, задачи вычисления собственных чисел и собственных векторов матриц. Эти задачи имеют очень важное теоретическое и прикладное значение. Трудности решения указанных задач, как правило, связаны с большой размерностью матриц.

Определение 1. Метод называется **точным**, если в предположении отсутствия ошибок округлений, получается точное решение за конечное число шагов.

Определение 2. Метод называется **итерационным**, если решение получается в виде предела элементов некоторой последовательности.

Метод исключения Гаусса (схема единственного деления) решения систем линейных алгебраических уравнений

Пусть дана система линейных алгебраических уравнений (СЛАУ)

$$A\vec{x} = \vec{b}, \quad (3.1)$$

где A – вещественная матрица порядка n , \vec{b} – заданный вектор, \vec{x} – искомый вектор.

Предположим, что $|A| \neq 0$. Тогда система (3.1) имеет единственное решение. Перепишем систему (3.1) в скалярном виде

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2, \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n. \end{cases} \quad (3.2)$$

Пусть $a_{11} \neq 0$. Разделив первое уравнение на a_{11} , имеем

$$x_1 + c_{12}x_2 + \dots + c_{1n}x_n = y_1, \quad (3.3)$$

$$\text{где } c_{1j} = \frac{a_{1j}}{a_{11}}, \quad j = 2, 3, \dots, n, \quad y_1 = \frac{b_1}{a_{11}}.$$

Теперь умножаем уравнение (3.3) последовательно на a_{i1} , $i = 2, 3, \dots, n$ и вычитаем, соответственно, из 2-го, 3-го, ..., n -го уравнений. В результате получаем

$$\left\{ \begin{array}{l} x_1 + c_{12}x_2 + c_{13}x_3 + \dots + c_{1n}x_n = y_1, \\ a_{22}^{(1)}x_2 + a_{23}^{(1)} + \dots + a_{2n}^{(1)}x_n = b_2^{(1)}, \\ a_{32}^{(1)}x_2 + a_{33}^{(1)} + \dots + a_{3n}^{(1)}x_n = b_3^{(1)}, \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ a_{n2}^{(1)}x_2 + a_{n3}^{(1)} + \dots + a_{nn}^{(1)}x_n = b_n^{(1)}. \end{array} \right. \quad (3.4)$$

Таким образом, x_1 исключили из всех уравнений начиная со второго.

Далее, первое уравнение системы (3.4) оставляем без изменения. Теперь, предполагая, что $a_{22}^{(1)} \neq 0$, делим на него второе уравнение в (3.4) и

аналогично предыдущему исключаем x_2 из всех уравнений, начиная с третьего, и т. д. В результате приходим к системе

$$\left\{ \begin{array}{l}
 x_1 + c_{12}x_2 + c_{13}x_3 + \dots + c_{1n}x_n = y_1, \\
 x_2 + c_{23}x_3 + \dots + c_{2n}x_n = y_2, \\
 \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\
 x_{n-1} + c_{n-1,n}x_n = y_{n-1}, \\
 x_n = y_n.
 \end{array} \right. \quad (3.5)$$

Матрица этой системы

$$C = \begin{pmatrix} 1 & c_{12} & c_{13} & \dots & c_{1,n-1} & c_{1n} \\ 0 & 1 & c_{23} & \dots & c_{2,n-1} & c_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & c_{n-1,n} \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix} \quad (3.6)$$

содержит нули всюду ниже главной диагонали. Такие матрицы называются *верхними треугольными*. Нетрудно проверить, что C^{-1} есть также верхняя треугольная матрица.

Преобразование системы (3.2) к эквивалентной системе (3.5) с верхней треугольной матрицей называется прямым ходом метода Гаусса. Вычисление неизвестных из (3.5) называется обратным ходом метода Гаусса.

Неизвестные x_i вычисляются так: из последнего уравнения находим x_n , из предпоследнего – x_{n-1} и т.д. Итак

$$x_i = y_i - \sum_{j=i+1}^n c_{ij} x_j, \quad i = n-1, \dots, 1; \quad x_n = y_n. \quad (3.7)$$

Число действий (трудоемкость метода). Для реализации метода Гаусса (схема единственного деления) требуется примерно $\frac{2}{3}n^3$ арифметических операций, причем подавляющее число этих действий совершается на этапе прямого хода.

Русские математики Ключев и Коковкин-Щербак доказали, что в общем случае система линейных алгебраических уравнений не может быть решена точными методами с помощью меньшего числа операций, чем требуется в Гауссовском исключении.

Метод Гаусса с выбором главного элемента по столбцу

Выбираем максимальный по модулю элемент в первом столбце матрицы A . Пусть $|a_{m1}| = \max_{1 \leq i \leq m} |a_{i1}|$, т.е. максимальный по модулю элемент первого столбца стоит в m -й строке. Меняем местами первое и m -е уравнения системы с соответствующей перенумерацией элементов этих уравнений. Делим новое первое уравнение на коэффициент при x_1 , имеем

$$\begin{aligned} x_1 + c_{12}x_2 + \dots + c_{1n}x_n &= y_1, \\ c_{1j} &= \frac{a_{1j}}{a_{11}}, \quad j = 2, 3, \dots, n, \quad y_1 = \frac{b_1}{a_{11}}, \end{aligned} \quad (3.8)$$

и, исключив x_1 из второго, третьего, ..., n -го уравнений точно так же, как в пункте 3.1, получаем систему

$$\left\{ \begin{array}{l} x_1 + c_{12}x_2 + c_{13}x_3 + \dots + c_{1n}x_n = y_1, \\ a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \dots + a_{2n}^{(1)}x_n = b_2^{(1)}, \\ a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 + \dots + a_{3n}^{(1)}x_n = b_3^{(1)}, \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ a_{n2}^{(1)}x_2 + a_{n3}^{(1)}x_3 + \dots + a_{nn}^{(1)}x_n = b_n^{(1)}. \end{array} \right. \quad (3.9)$$

Далее первое уравнение системы (3.9) оставляем без изменения. Выбираем максимальный по модулю элемент во втором столбце матрицы системы (3.9), начиная со второй строки. Пусть $|a_{l2}^{(1)}| = \max_{2 \leq i \leq n} |a_{i2}^{(1)}|$, т.е. максимальный по модулю элемент второго укороченного столбца стоит в l -той строке. Меняем местами второе и l -тое уравнения системы (3.9). Исключаем x_2 из всех уравнений, начиная с третьего, и т.д. После n шагов получаем систему уравнений с верхней треугольной матрицей.

Обратный ход метода Гаусса (т.е. вычисление неизвестных x_i , $i = 1, \dots, n$) осуществляем так же, как и в схеме единственного деления.

Замечание. Метод Гаусса с частичным выбором главного элемента можно применять к любой СЛАУ с невырожденной матрицей. Однако этот метод не всегда вычислительно устойчив. Например, пусть методом Гаусса с выбором главного элемента по столбцу решается система $A\vec{x} = \vec{b}$ с матрицей коэффициентов

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 \\ -1 & 1 & 0 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & 1 \end{pmatrix}.$$

Приведение такой матрицы к треугольному виду прямым ходом метода Гаусса равносильно следующей последовательности эквивалентных преобразований матрицы A :

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 \\ -1 & 1 & 0 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & -1 & 1 & 2 \\ 0 & -1 & -1 & 2 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & -1 & 4 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 8 \end{pmatrix}.$$

Компактная схема Гаусса

Пусть осуществлены первые $(k-1)$ шагов, т.е. уже исключены x_1, \dots, x_{k-1} , следовательно, система (3.1) приведена к виду:

$$\left\{ \begin{array}{l} x_1 + c_{12}x_2 + \dots + c_{1k}x_k + \dots + c_{1n}x_n = y_1, \\ \quad x_2 + \dots + c_{2k}x_k + \dots + c_{2n}x_n = y_2, \\ \quad \quad \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ \quad \quad \quad x_{k-1} + c_{k-1,k}x_k + \dots + c_{k-1,n}x_n = y_{k-1}, \\ \quad \quad \quad \quad a_{kk}^{(k-1)}x_k + \dots + a_{kn}^{(k-1)}x_n = b_k^{(k-1)}, \\ \quad \quad \quad \quad \quad \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ \quad \quad \quad \quad \quad \quad a_{nk}^{(k-1)}x_k + \dots + a_{nn}^{(k-1)}x_n = b_n^{(k-1)}. \end{array} \right. \quad (3.10)$$

Рассмотрим k -е уравнение системы (3.10) и предположим, что $a_{kk}^{(k-1)} \neq 0$. Разделив обе части этого уравнения на $a_{kk}^{(k-1)}$, получаем

$$x_k + c_{k,k+1}x_{k+1} + \dots + c_{k,n}x_n = y_k, \quad (3.11)$$

где

$$c_{kj} = \frac{a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}}, \quad j = k+1, \dots, n, \quad y_k = \frac{b_k^{(k-1)}}{a_{kk}^{(k-1)}}.$$

Теперь уравнение (3.11) умножаем последовательно на $a_{ik}^{(k-1)}$ и вычитаем, соответственно, из (i -го) уравнения системы (3.10). В результате последняя группа уравнений системы (3.10) принимает вид

$$\left\{ \begin{array}{l} x_k + c_{k,k+1}x_{k+1} + \dots + c_{k,n}x_n = y_k, \\ a_{k+1,k+1}^{(k)}x_{k+1} + \dots + a_{k+1,n}^{(k)}x_n = b_{k+1}^{(k)}, \\ \dots \dots \dots \dots \dots \dots \dots \dots \\ a_{n,k+1}^{(k)}x_{k+1} + \dots + a_{nn}^{(k)}x_n = b_n^{(k)}, \end{array} \right. \quad (3.12)$$

где

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - a_{ik}^{(k-1)}c_{kj}, \quad i, j = k+1, \dots, n; \quad b_i^{(k)} = b_i^{(k-1)} - a_{ik}^{(k-1)}y_k, \quad i = k+1, \dots, n.$$

Таким образом, в прямом ходе метода Гаусса коэффициенты уравнений преобразуются по формулам

$$a_{kj}^{(0)} = a_{kj}, \quad k, j = 1, 2, \dots, n;$$
$$c_{kj} = \frac{a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}}, \quad j = k+1, \dots, n, \quad k = 1, 2, \dots, n; \quad (3.13)$$

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - a_{ik}^{(k-1)} c_{kj}, \quad i, j = k+1, \dots, n, \quad k = 1, \dots, n. \quad (3.14)$$

Вычисление правых частей системы (3.5) осуществляется по формулам

$$b_k^{(0)} = b_k, \quad y_k = \frac{b_k^{(k-1)}}{a_{kk}^{(k-1)}}, \quad k = 1, \dots, n; \quad (3.15)$$

$$b_i^{(k)} = b_i^{(k-1)} - a_{ik}^{(k-1)} y_k, \quad i = k+1, \dots, n. \quad (3.16)$$

По формулам (3.13) – (3.16) вычисляем коэффициенты c_{ij} , правые части y_i , $i = 1, \dots, n$, $j = i+1, \dots, n$, системы (3.5) и затем осуществляем обратный ход, как и в схеме единственного деления.

После первого шага система принимает вид:

$$\begin{cases} x_1 & + & a_{13}^{(2)}x_3 & + \dots & + & a_{1n}^{(2)}x_n & = & b_1^{(2)}, \\ & x_2 & + & a_{23}^{(2)}x_3 & + \dots & + & a_{2n}^{(2)}x_n & = & b_2^{(2)}, \\ a_{31}x_1 & + & a_{32}x_2 & + & a_{33}x_3 & + \dots & + & a_{3n}x_n & = & b_3, \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1}x_1 & + & a_{n2}x_2 & + & a_{n3}x_3 & + \dots & + & a_{nn}x_n & = & b_n. \end{cases}$$

Теперь исключаем x_1 и x_2 из третьего уравнения. Преобразованное третье уравнение делим на коэффициент при x_3 и исключаем x_3 из первых двух уравнений, и т.д. В итоге получаем $x_i = y_i$, $i = 1, 2, \dots, n$, т.е. мы вычислили решение системы (3.17).

LU-разложение

$$A = LU,$$

L – нижняя треугольная матрица, а U – верхняя.

решение системы $A\vec{x} = \vec{b}$ сводится к

решению двух систем с треугольными матрицами $L\vec{y} = \vec{b}$ и $U\vec{x} = \vec{y}$.

Теорема об LU-разложении. Пусть все главные миноры матрицы A отличны от нуля ($\Delta_j \neq 0, j = 1, \dots, n$), тогда матрицу A можно представить единственным образом в виде произведения $A = LU$, где L – нижняя треугольная матрица с ненулевыми диагональными элементами, U – верхняя треугольная матрица с единицами по главной диагонали.

Если все главные миноры матрицы A отличны от нуля, то справедливы рекуррентные формулы, позволяющие найти элементы матриц L и U :

$$u_{11} = a_{11},$$

$$u_{1j} = a_{1j}, \quad l_{j1} = \frac{a_{j1}}{u_{11}}, \quad j = 2, 3, \dots, n,$$

$$u_{ii} = a_{ii} - \sum_{p=1}^{i-1} l_{ip} u_{pi}, \quad i = 2, 3, \dots, n,$$

$$u_{ij} = a_{ij} - \sum_{p=1}^{i-1} l_{ip} u_{pj}, \quad l_{ji} = \frac{a_{ji} - \sum_{p=1}^{i-1} l_{jp} u_{pi}}{u_{ii}},$$

$i = 2, 3, \dots, n, \quad j = i + 1, i + 2, \dots, n$. Здесь $l_{ii} = 1$.

Метод квадратных корней

Пусть $A = (a_{ij})_{i,j=1}^n$ – данная симметрическая матрица, т.е. $a_{ij} = a_{ji}$.

Представим ее в виде $A = U^T U$, где

$$U = \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & u_{nn} \end{pmatrix}, \quad U^T = \begin{pmatrix} u_{11} & 0 & \dots & 0 \\ u_{12} & u_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ u_{1n} & u_{2n} & \dots & u_{nn} \end{pmatrix}.$$

Составим систему $\frac{n(n+1)}{2}$ уравнений относительно такого же количества неизвестных (элементов матрицы U):

$$\begin{array}{rcccc}
 u_{11}^2 = a_{11}, & u_{12}u_{11} = a_{12}, & \dots, & u_{11}u_{1n} = a_{1n}, \\
 & u_{12}^2 + u_{22}^2 = a_{22}, & \dots, & u_{12}u_{1n} + u_{22}u_{2n} = a_{2n}, \\
 & & & \dots \\
 & & & u_{1n}^2 + u_{2n}^2 + \dots + u_{nn}^2 = a_{nn}.
 \end{array}$$

Из первой строки уравнений находим сначала $u_{11} = \sqrt{a_{11}}$, затем

$$u_{1j} = \frac{a_{1j}}{u_{11}} \text{ при } j = 2, \dots, n. \text{ Из второй} - u_{22} = \sqrt{a_{22} - u_{12}^2}, \text{ затем } u_{2j} = \frac{a_{2j} - u_{12}u_{1j}}{u_{22}}$$

при $j = 3, \dots, n$ и т.д. Завершается процесс вычислением $u_{nn} = \sqrt{a_{nn} - \sum_{k=1}^{n-1} u_{kn}^2}$.

Таким образом, матрица U может быть определена совокупностью формул

$$u_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} u_{ki}^2} \text{ при } i = 1, 2, \dots, n;$$

$$u_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} u_{ki}u_{kj}}{u_{ii}} \text{ при } i = 2, \dots, n, j > i \text{ (} u_{ij} = 0 \text{ при } j < i \text{)}. \quad (3.24)$$

При наличии $U^T U$ -разложения решение системы с симметрической матрицей $A\vec{x} = \vec{b}$ сводится к последовательному решению двух систем с треугольными матрицами $U^T \vec{y} = \vec{b}$ и $U\vec{x} = \vec{y}$.

Первая из них имеет вид

$$\begin{cases} u_{11}y_1 & = b_1, \\ u_{12}y_1 + u_{22}y_2 & = b_2, \\ \dots & \dots \\ u_{1n}y_1 + u_{2n}y_2 + \dots + u_{nn}y_n & = b_n, \end{cases}$$

откуда получаем вспомогательные неизвестные y_1, y_2, \dots, y_n по формулам

$$y_1 = \frac{b_1}{u_{11}}, \quad y_i = \frac{b_i - \sum_{k=1}^{i-1} u_{ki}y_k}{u_{ii}} \quad ((i > 1)). \quad (3.25)$$

Из второй системы

$$\left\{ \begin{array}{l} u_{11}x_1 + u_{12}x_2 + \dots + u_{1n}x_n = y_1, \\ u_{22}x_2 + u_{2n}x_n = y_2, \\ \dots \\ u_{nn}x_n = y_n \end{array} \right.$$

находим искомые значения x_i в обратном порядке, т.е. при $i = n, n-1, \dots, 1$, по формулам

$$x_n = \frac{y_n}{u_{nn}}, \quad x_i = \frac{y_i - \sum_{k=i+1}^n u_{ik}x_k}{u_{ii}} \quad (i < n). \quad (3.26)$$

Метод вращений

Пусть c_{12} и s_{12} – некоторые отличные от нуля числа. Умножим первое уравнение системы

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2, \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases} \quad (3.27)$$

на c_{12} , второе – на s_{12} и сложим их. Полученным уравнением заменяем первое уравнение системы. Затем первое уравнение исходной системы умножаем на $-s_{12}$, второе – на c_{12} и результатом их сложения заменяем второе уравнение. Таким образом, первые два уравнения системы (3.27) заменяются уравнениями

$$\begin{aligned}
 (c_{12}a_{11} + s_{12}a_{21})x_1 + (c_{12}a_{12} + s_{12}a_{22})x_2 + \dots + (c_{12}a_{1n} + s_{12}a_{2n})x_n &= c_{12}b_1 + s_{12}b_2, \\
 (-s_{12}a_{11} + c_{12}a_{21})x_1 + (-s_{12}a_{12} + c_{12}a_{22})x_2 + \dots + (-s_{12}a_{1n} + c_{12}a_{2n})x_n &= -s_{12}b_1 + c_{12}b_2.
 \end{aligned}$$

На введенные два параметра c_{12} и s_{12} наложим два условия:

$-s_{12}a_{11} + c_{12}a_{21} = 0$ – условие обнуления (т.е. исключения x_1 из второго уравнения);

$c_{12}^2 + s_{12}^2 = 1$ – условие нормировки.

Легко проверить, что за c_{12} и s_{12} , удовлетворяющие этим условиям, можно принять, соответственно,

$$c_{12} = \frac{a_{11}}{\sqrt{a_{11}^2 + a_{21}^2}}, \quad s_{12} = \frac{a_{21}}{\sqrt{a_{11}^2 + a_{21}^2}}. \quad (3.28)$$

После проведенных преобразований система (3.27) принимает вид

$$\begin{cases} a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + \dots + a_{1n}^{(1)}x_n = b_1^{(1)}, \\ a_{22}^{(1)}x_2 + \dots + a_{2n}^{(1)}x_n = b_2^{(1)}, \\ a_{31}x_1 + a_{32}x_2 + \dots + a_{3n}x_n = b_3, \\ \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n, \end{cases} \quad (3.29)$$

где

$$a_{1j}^{(1)} = c_{12}a_{1j} + s_{12}a_{2j}, \quad j = 1, \dots, n, \quad b_1^{(1)} = c_{12}b_1 + s_{12}b_2,$$
$$a_{2j}^{(1)} = -s_{12}a_{1j} + c_{12}a_{2j}, \quad j = 2, 3, \dots, n, \quad b_2^{(1)} = -s_{12}b_1 + c_{12}b_2.$$

Далее первое уравнение системы (3.29) заменяем новым, полученным сложением результатов умножения первого и третьего уравнений (3.29),

соответственно, на $c_{13} = \frac{a_{11}^{(1)}}{\sqrt{(a_{11}^{(1)})^2 + a_{31}^2}}$ и $s_{13} = \frac{a_{31}}{\sqrt{(a_{11}^{(1)})^2 + a_{31}^2}}$, а третье заменяем

уравнением, полученным сложением результатов умножения тех же уравнений, соответственно, на $-s_{13}$ и c_{13} . Получаем систему

$$\left\{ \begin{array}{l} a_{11}^{(2)}x_1 + a_{12}^{(2)}x_2 + \dots + a_{1n}^{(2)}x_n = b_1^{(2)}, \\ a_{22}^{(1)}x_2 + \dots + a_{2n}^{(1)}x_n = b_2^{(1)}, \\ a_{32}^{(1)}x_2 + \dots + a_{3n}^{(1)}x_n = b_3^{(1)}, \\ a_{41}x_1 + a_{42}x_2 + \dots + a_{4n}x_n = b_4, \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n, \end{array} \right.$$

где

$$\begin{aligned} a_{1j}^{(2)} &= c_{13}a_{1j}^{(1)} + s_{13}a_{3j}, \quad j = 1, \dots, n, \quad b_1^{(2)} = c_{13}b_1^{(1)} + s_{13}b_3, \\ a_{3j}^{(1)} &= -s_{13}a_{1j}^{(1)} + c_{13}a_{3j}, \quad j = 2, 3, \dots, n, \quad b_3^{(1)} = -s_{13}b_1^{(1)} + c_{13}b_3. \end{aligned}$$

Проделав такие преобразования $(n-1)$ раз, приходим к системе

$$\left\{ \begin{array}{l} a_{11}^{(n-1)}x_1 + a_{12}^{(n-1)}x_2 + \dots + a_{1n}^{(n-1)}x_n = b_1^{(n-1)}, \\ \quad \quad \quad a_{22}^{(1)}x_2 + \dots + a_{2n}^{(1)}x_n = b_2^{(1)}, \\ \quad \quad \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ \quad \quad \quad a_{n2}^{(1)}x_2 + \dots + a_{nn}^{(1)}x_n = b_n^{(1)} \end{array} \right. \quad (3.30)$$

$$\left\{ \begin{array}{l}
 x_1 + c_{12}x_2 + c_{13}x_3 + \dots + c_{1n}x_n = y_1, \\
 \tilde{a}_{22}^{(1)}x_2 + \tilde{a}_{23}^{(1)} + \dots + \tilde{a}_{2n}^{(1)}x_n = \tilde{b}_2^{(1)}, \\
 \tilde{a}_{32}^{(1)}x_2 + \tilde{a}_{33}^{(1)} + \dots + \tilde{a}_{3n}^{(1)}x_n = \tilde{b}_3^{(1)}, \\
 \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\
 \tilde{a}_{n2}^{(1)}x_2 + \tilde{a}_{n3}^{(1)} + \dots + \tilde{a}_{nn}^{(1)}x_n = \tilde{b}_n^{(1)}.
 \end{array} \right. \quad (3.31)$$

Дальше точно так же за $(n-2)$ промежуточных шага преобразуем подсистему

$$\begin{cases} a_{22}^{(1)}x_2 + \dots + a_{2n}^{(1)}x_n = b_2^{(1)}, \\ \dots \dots \dots \dots \dots \dots \dots \\ a_{n2}^{(1)}x_2 + \dots + a_{nn}^{(1)}x_n = b_n^{(1)}, \end{cases}$$

системы (3.30), создавая нули под элементом $a_{22}^{(1)}$ и т.д.

В результате $(n-1)$ таких этапов прямого хода исходная система (3.27) будет приведена к треугольному виду

$$\left\{ \begin{array}{l} a_{11}^{(n-1)}x_1 + a_{12}^{(n-1)}x_2 + \dots + a_{1n}^{(n-1)}x_n = b_1^{(n-1)}, \\ a_{22}^{(n-1)}x_2 + \dots + a_{2n}^{(n-1)}x_n = b_2^{(n-1)}, \\ \dots \quad \dots \quad \dots \quad \dots \\ a_{nn}^{(n-1)}x_n = b_n^{(n-1)}. \end{array} \right.$$

Нахождение отсюда неизвестных x_n, x_{n-1}, \dots, x_1 не отличается от рассмотренного ранее обратного хода метода Гаусса.

Векторные и матричные нормы. Согласованность норм. Обусловленность СЛАУ. Число обусловленности матрицы. Вычисление определителей. Обращение матриц.

Пусть N – линейное пространство n -мерных векторов.

Напомним, что в N задана норма, если каждому вектору \vec{x} из N сопоставлено вещественное число $\|\vec{x}\|$, удовлетворяющее аксиомам:

1. $\|\vec{x}\| \geq 0$ и $\|\vec{x}\| = 0$ тогда и только тогда, когда $\vec{x} = \vec{0}$.
2. $\|\alpha\vec{x}\| = |\alpha| \cdot \|\vec{x}\|$ для любого числа α и любого $\vec{x} \in N$.
3. $\|\vec{x} + \vec{y}\| \leq \|\vec{x}\| + \|\vec{y}\|$.

Аналогично определяется норма матриц. Напомним, что норма вектора может быть вычислена, в частности, по одной из следующих формул:

$$\|\vec{x}\|_1 = \sum_{i=1}^n |x_i|, \quad \|\vec{x}\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}, \quad \|\vec{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|,$$

а норма матрицы по формулам:

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|, \quad \|A\|_2 = \sqrt{\max_{1 \leq j \leq n} |\lambda_j(A^T \cdot A)|}, \quad \|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|.$$

Определение 1. Данная норма матриц называется **согласованной** с данной нормой векторов, если для любой матрицы A и для любого вектора \vec{x} справедливо неравенство

$$\|A\vec{x}\| \leq \|A\| \cdot \|\vec{x}\|.$$

Нетрудно проверить, что приведенные нормы согласованы. В дальнейшем, не оговаривая особо, будем считать, что нормы матриц и векторов согласованы.

Определение 2. Вектор $\Delta\vec{x}_\varepsilon = \vec{x}_\varepsilon - \vec{x}$ называется **вектором погрешности**. Здесь через \vec{x} обозначили точное решение системы $A\vec{x} = \vec{b}$, а через \vec{x}_ε – результат вычислений (говорят « x вычисленное»).

Определение 3. Вектор $\vec{r} = A\vec{x}_\varepsilon - \vec{b}$ называется **вектором невязки**.

Итак, малость вектора невязки необходима для того, чтобы вычисленные значения неизвестных были близки к точным, но не достаточна. Докажем последнее утверждение. Действительно, так как

$$\vec{r} = A\vec{x}_a - \vec{b} = A(\vec{x}_a - \vec{x}) = A\Delta\vec{x}_a,$$

то

$$\Delta\vec{x}_a = A^{-1}\vec{r} \text{ и } \|\Delta\vec{x}_a\| \leq \|A^{-1}\| \cdot \|\vec{r}\|.$$

Из последнего неравенства вытекает, что из малости $\|\vec{r}\|$ не обязательно следует малость $\|\Delta\vec{x}_a\|$, т.е. близость вычисленного решения к точному решению системы $A\vec{x} = \vec{b}$.

Видно, что $\|\Delta\vec{x}_a\|$ может быть большой, если велика норма обратной матрицы A^{-1} . Последнее наблюдается у так называемых плохо обусловленных матриц.

Под обусловленностью вычислительной задачи понимают чувствительность ее решения к малым погрешностям входных данных.

Задачу называют хорошо обусловленной, если малым погрешностям входных данных отвечают малые погрешности решения, и плохо обусловленной, если возможны большие изменения решения.

Часто оказывается возможным ввести количественную меру степени обусловленности вычислительной задачи – число обусловленности. Эту величину можно интерпретировать как коэффициент возможного возрастания погрешностей в решении по отношению к вызвавшим их погрешностям входных данных.

Пусть между погрешностями входных данных \tilde{x} и решениями \tilde{y} установлены неравенства $\Delta(\tilde{y}) \leq v_{\Delta} \Delta(\tilde{x})$, $\delta(\tilde{y}) \leq v_{\delta} \delta(\tilde{x})$.

Величины v_{Δ} и v_{δ} называют, соответственно, абсолютным и относительным числом обусловленности. Чаще всего под числом обусловленности v задачи понимают относительное число обусловленности.

Предположим, что в системе $A\vec{x} = \vec{b}$ возмущены (допущены ошибки при измерениях, произведены округления при вычислениях) как правая часть, так и коэффициенты. Рассмотрим возмущённую систему $\tilde{A}\tilde{x} = \tilde{b}$ и обозначим $\Delta A = \tilde{A} - A$, $\Delta\vec{x} = \tilde{x} - \vec{x}$, $\Delta\vec{b} = \tilde{b} - \vec{b}$. Если A имеет обратную матрицу A^{-1} и выполнено условие $\|\Delta A\| < \frac{1}{\|A^{-1}\|}$, то матрица $\tilde{A} = A + \Delta A$ имеет обратную и справедливы оценки относительной погрешности

$$\frac{\|\Delta \vec{x}\|}{\|\vec{x}\|} \leq \frac{\text{cond}(A)}{1 - \text{cond}(A)(\|A^{-1}\|/\|A\|)} \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta \vec{b}\|}{\|\vec{b}\|} \right),$$

$$\|\delta(\tilde{x})\| \leq \|A\| \|A^{-1}\| \|\delta(\tilde{A})\|,$$

$$\|\delta(\tilde{x})\| \leq \|A\| \|A^{-1}\| \|\delta(\tilde{b})\|,$$

где через $\text{cond}(A)$ обозначили произведение $\|A\| \cdot \|A^{-1}\|$.

Определение 4. Число $\text{cond}(A) = \|A\| \cdot \|A^{-1}\|$ называют числом обусловленности матрицы, а вернее, стандартным числом обусловленности матрицы.

Определение 5. Систему линейных алгебраических уравнений (матрицу A) называют плохо обусловленной, если число обусловленности велико, в противном случае – хорошо обусловленной.

Свойства $\text{cond}(A)$:

1. $\text{cond}(E) = 1$.

2. $\text{cond}(A) \geq 1$.

3. $\text{cond}(AB) \leq \text{cond}(A)\text{cond}(B)$.

4. Число обусловленности не меняется при умножении матрицы A на ненулевое число.

5. Для симметрической матрицы A :
$$\text{cond}(A) = \frac{\max_{1 \leq j \leq n} |\lambda_j(A)|}{\min_{1 \leq j \leq n} |\lambda_j(A)|}.$$

Традиционным примером очень плохо обусловленной матрицы является матрица Гильберта A с элементами $a_{ij} = \frac{1}{i+j-1}$. Из табл. 1 видно, что для матрицы A даже сравнительно невысокого порядка число обусловленности оказывается чрезвычайно большим.

Таблица 4.1

Порядок матрицы Гильберта	2	3	4	5	6	7	8	9	10
Приближенное значение $cond(A)$	$2 \cdot 10^1$	$5 \cdot 10^2$	$2 \cdot 10^4$	$5 \cdot 10^5$	$2 \cdot 10^7$	$5 \cdot 10^8$	$2 \cdot 10^{10}$	$5 \cdot 10^{11}$	$2 \cdot 10^{13}$

Основная задача вычислительной алгебры – решение систем линейных алгебраических уравнений (СЛАУ):

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1,$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2,$$

...

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n.$$

Предполагается, что матрица A неособенная, т. е. $\det A \neq 0$, и решение единственно.

Метод простой итерации (метод Якоби).

$$Ax = b; x, b \in R^n.$$

Представим $A = A^- + D + A^+$, где D – диагональная матрица с диагональными членами матрицы A ; A^- – часть матрицы A , лежащая ниже центральной диагонали; A^+ – часть матрицы A , лежащая выше центральной диагонали. Тогда

$$(A^- + D + A^+)x = b,$$

или

$$Dx = -(A^- + A^+)x + b.$$

Запишем итерационный метод в виде

$$Dx^{k+1} = -(A^- + A^+)x^k + b; k = 0, 1, 2, \dots$$

Разрешим его относительно x^{k+1} :

$$x^{k+1} = -D^{-1}(A^- + A^+)x^k + D^{-1}b; k = 0, 1, \dots$$

$$x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^k - \sum_{j=i+1}^n a_{ij} x_j^k \right); i = \overline{1, n}.$$

Для реализации метода простой итерации должны быть заданы матрица СЛАУ A , вектор свободных членов B , начальное приближение вектора X , точность вычислений ε . Тогда новые значения вектора X вычисляются по формуле (10.1).

Критерий завершения процесса вычислений

$$\|x^{k+1} - x^k\| = \max |x_i^{k+1} - x_i^k| < \varepsilon, 1 \leq i \leq n,$$

где x^k – приближенное значение решения на k -м шаге численного метода.

Процедура вычисления значений вектора по методу простой итерации

```
int ind(int i,int j,int SIZE)    /* процедура обращения к элементу матрицы */
{    return (i*(SIZE+1)+j); }

void Iter_Jacoby(double *A, double *X, double *X_old, int size)
    /* задана матрица A, начальное приближение вектора X_old,
    размерность матрицы size, вычисляем новое значение вектора X */
{
    unsigned int i, j;
    double Sum;
    for (i = 0; i < size; ++i)
    {
        Sum = 0;
        for (j = 0; j < i; ++j)
            Sum += A[ind(i,j,size)] * X_old[j];
        for (j = i+1; j < size; ++j)
            Sum += A[ind(i,j,size)] * X_old[j];
        X[i]=(A[ind(i,size,size)] - Sum) / A[ind(i,i,size)];
    }
}
```

```

unsigned long int SolveSLAE(double *A, double *X, double Error, int size)
/* задана матрица A размерности size+1 (матрица+столбец свободных
   членов), начальное приближение вектора X, погрешность вычислений Error */
{ double X_old; /* предыдущее значение вектора X */
  int i, Iter = 0; /* число итераций */
  double dNorm, dVal;
  X_old = malloc(sizeof(double) * size); /* выделяем память для X_old */
  do{
    ++Iter;
    /*сохраняем предыдущее значение вектора X */
    memcpy(X_old, X, size);
    /* вычисляем новое значение вектора X */
    Iter_Jacoby(A, X, X_old, size);
    dNorm = 0; /* считаем норму погрешности */
    for (i = 0; i < size; ++i)
    {
      dVal = fabs(X[i] - X_old[i]);
      if (dNorm < dVal) dNorm = dVal;
    }
  }while(Error < dNorm); /* цикл до достижения необходимой точности */
  free(X_old);
  return Iter;
}

```

Метод Гаусса–Зейделя.

$$(A^- + D + A^+)x = b.$$

Итерационная схема Гаусса–Зейделя следует из этого представления системы:

$$(A^- + D)x^{k+1} = b - A^+x^k ; k = 0,1,2,\dots,$$

или

$$Dx^{k+1} = -A^-x^{k+1} - A^+x^k + b ; k = 0,1,2,\dots$$

$$x^{k+1} = -(A^- + D)^{-1} A^+ x^k + (A^- + D)^{-1} b; \quad k = 0, 1, \dots$$

$$B = -(A^- + D)^{-1} A^+ = -(A^- + D)^{-1} (A - D - A^-) = I - (A^- + D)^{-1} A.$$

$$x_i^{k+1} = \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^n a_{ij} x_j^k \right) / a_{ii}; \quad i = \overline{1, n}; \quad k = 0, 1, 2, \dots$$

```
void GaussZeidel (double *A, double *X, int size)
```

```
    /* задана матрица A, начальное приближение вектора X,
```

```
       размерность матрицы size, вычисляем новое значение вектора X */
```

```
{ unsigned int i, j;
```

```
  double Sum;
```

```
  for (i = 0; i < size; ++i) {
```

```
    Sum = 0;
```

```
    for (j = 0; j < i; ++j)
```

```
      Sum += A[ind(i,j,size)] * X[j];
```

```
    for (j = i+1; j < size; ++j)
```

```
      Sum += A[ind(i,j,size)] * X[j];
```

```
    X[i]=(A[ind(i,size,size)] - Sum) / A[ind(i,i,size)];
```

```
  }
```

```
}
```