

СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ

После изучения данной темы вы сможете: проводить численное решение задач линейной алгебры.

К решению систем линейных уравнений сводятся многочисленные практические задачи, решение линейных систем является одной из самых распространенных и важных задач вычислительной математики.

Запишем систему n линейных алгебраических уравнений с n неизвестными:

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

Систему уравнений (1) можно записать в векторно-матричном виде:

$$A\vec{x} = \vec{b} \quad (2)$$

где A – матрица коэффициентов системы, \vec{x} и \vec{b} – вектор-столбец неизвестных и вектор-столбец правых частей соответственно:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}, \quad \vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad \vec{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

или, в более компактной записи $\vec{x} = \{x_1, x_2, \dots, x_n\}$, $\vec{b} = \{b_1, b_2, \dots, b_n\}$.

Определителем (детерминантом) матрицы A порядка n называется число D , равное

$$D = \det A = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} = \sum (-1)^k a_{1\alpha} a_{2\beta} \dots a_{n\varpi} \quad (3)$$

Здесь индексы $\alpha, \beta, \dots, \varpi$ пробегает все возможные $n!$ перестановок номеров $1, 2, \dots, n$; k – число инверсий в данной перестановке (здесь инверсия – обмен двух индексов местами; с помощью таких обменов перестановка $\alpha, \beta, \dots, \varpi$ получается из перестановки $1, 2, \dots, n$).

Необходимым и достаточным условием существования единственного решения системы линейных уравнений является условие $D \neq 0$. В случае равенства нулю определителя системы матрица называется *вырожденной*; при этом система линейных уравнений (1) либо не имеет решения, либо имеет их бесконечное множество. При вычислениях на компьютере, когда происходят округление или отбрасывание младших разрядов чисел, далеко не всегда удается получить точное равенство определителя нулю. Системы уравнений для которых $D \approx 0$ называются *плохо обусловленными*.

Методы решения систем линейных уравнений. Методы решения линейных систем делятся на две группы – прямые и итерационные. *Прямые методы* используют конечные соотношения (формулы) для вычисления неизвестных. Они дают решение после выполнения заранее известного числа операций.

Преимущества	Недостатки
<ul style="list-style-type: none"> методы сравнительно просты и наиболее универсальны (т. е. пригодны для решения широкого класса линейных систем). 	<ul style="list-style-type: none"> как правило, требуют хранения в оперативной памяти компьютера сразу всей матрицы (при больших n расходуется много места в памяти); обычно не учитывают структуру матрицы при большом числе нулевых элементов в

разреженных матрицах (например, клеточных или ленточных) эти элементы занимают место в памяти машины, и над ними проводятся арифметические действия;

- накопление погрешностей в процессе решения, поскольку вычисления на любом этапе используют результаты предыдущих операций (особенно опасно для больших систем, когда резко возрастает общее число операций, а также для плохо обусловленных систем, весьма чувствительных к погрешностям).

Прямые методы решения линейных систем иногда называют *точными*, поскольку решение выражается в виде точных формул через коэффициенты системы. Однако точное решение может быть получено лишь при точном выполнении вычислений и при точных значениях коэффициентов системы. На практике же при использовании компьютеров вычисления проводятся с погрешностями. Поэтому неизбежны погрешности и в окончательных результатах. Прямые методы используются обычно для не слишком больших ($n < 1000$) систем с плотно заполненной матрицей и не близким к нулю определителем.

Итерационные методы – это методы последовательных приближений. В них необходимо задать некоторое приближенное решение – *начальное приближение*. После этого с помощью некоторого алгоритма проводится один цикл вычислений, называемый *итерацией*. В результате итерации находят новое приближение. Итерации проводятся до получения решения с требуемой точностью.

Преимущества	Недостатки
<ul style="list-style-type: none"> • требуют хранения в памяти машины не всей матрицы системы, а лишь нескольких векторов с n компонентами; • погрешности окончательных результатов при использовании итерационных методов не накапливаются, поскольку точность вычислений в каждой итерации определяется лишь результатами предыдущей итерации и практически не зависит от ранее выполненных вычислений. 	<ul style="list-style-type: none"> • алгоритмы решения обычно более сложные по сравнению с прямыми методами; • объем вычислений заранее определить трудно; • сходимость итераций может быть очень медленной (ищутся эффективные пути ее ускорения).

Итерационные методы в ряде случаев предпочтительнее, например, они особенно полезны в случае большого числа уравнений, а также плохо обусловленных систем.

Итерационные методы могут использоваться для уточнения решений, полученных с помощью прямых методов. Такие смешанные алгоритмы обычно довольно эффективны, особенно для плохо обусловленных систем. В последнем случае могут также применяться методы регуляризации.

Другие задачи линейной алгебры. Кроме решения систем линейных уравнений существуют другие задачи линейной алгебры – вычисление определителя, обратной матрицы, собственных значений матрицы и др.

Легко вычисляются лишь определители невысоких порядков и некоторые специальные типы определителей. В частности, для определителей второго и третьего порядков соответственно имеем

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21},$$

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{21}a_{32}a_{13} -$$

$$- a_{31}a_{22}a_{13} - a_{21}a_{12}a_{33} - a_{32}a_{23}a_{11}.$$

В общем случае вычисление определителя оказывается значительно более трудоемким. Определитель D порядка n имеет вид

$$D = \sum (-1)^k a_{1\alpha} a_{2\beta} \dots a_{n\omega}.$$

Для вычисления определителя порядка n (без использования специальных приемов) требуется $(n - 1)n!$ умножений и $n! - 1$ сложений, т. е. общее число арифметических операций равно

$$N = n \times n! - 1 \approx n \times n!. \quad (4)$$

Можно подсчитать время вычисления таких определителей на компьютере с быстродействием равным 10 млн. операций в секунду. Так для вычисления определителя 10-го порядка потребуется около 3,6 с, а при $n = 20$ – свыше 150 тыс. лет. Следовательно, необходимо разрабатывать и использовать экономичные численные методы вычисления определителей.

Прямые методы

Одним из способов решения системы линейных уравнений является *правило Крамера*, согласно которому каждое неизвестное представляется в виде отношения определителей. Запишем его для системы

$$\begin{aligned} a_1x + b_1y &= c_1, \\ a_2x + b_2y &= c_2. \end{aligned}$$

Тогда

$$x = D_1/D, \quad y = D_2/D,$$

$$D = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}, \quad D_1 = \begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}, \quad D_2 = \begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}$$

Можно использовать это правило для решения систем уравнений произвольного порядка. Однако при большом числе уравнений потребуется выполнить огромное число арифметических операций, поскольку для вычислений n неизвестных необходимо найти значения определителей, число которых $n + 1$. Поэтому правило Крамера можно использовать лишь для решения систем, состоящих из нескольких уравнений.

Другой метод решения линейной системы основан на использовании обратной матрицы.

Система записывается в виде $A\vec{x} = \vec{b}$. Умножая обе части этого векторного уравнения слева на обратную матрицу A^{-1} , получаем $\vec{x} = A^{-1}\vec{b}$. Однако если не использовать экономичных схем для вычисления обратной матрицы, этот способ также непригоден для практического решения линейных систем при больших значениях n из-за большого объема вычислений.

Наиболее распространенными среди прямых методов являются *метод исключения Гаусса* и его модификации.

Метод (исключения) Гаусса. Он основан на приведении матрицы системы к треугольному виду. Это достигается последовательным исключением неизвестных из уравнений системы. Сначала с помощью первого уравнения исключается x_1 из всех последующих уравнений системы. Затем с помощью второго уравнения исключается x_2 из третьего и всех последующих уравнений. Этот процесс, называемый *прямым ходом метода Гаусса*, продолжается до тех пор, пока в левой части последнего (n -го) уравнения не останется лишь один член с неизвестным x_n , т. е. матрица системы будет приведена к треугольному виду.

Обратный ход метода Гаусса состоит в последовательном вычислении искомого неизвестных: решая последнее уравнение, находим единственное в этом уравнении неизвестное x_n . Далее, используя это значение, из предыдущего уравнения вычисляем x_{n-1} и т. д. Последним найдем x_1 из первого уравнения.

Описанные процедуры применимы лишь для систем с невырожденной матрицей. В противном случае с помощью метода Гаусса можно ответить на вопрос, имеет ли система бесконечное множество решений или не имеет ни одного.

Рассмотрим применение метода Гаусса для системы

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3. \end{aligned} \quad (5)$$

Для исключения x_1 из второго уравнения прибавим к нему первое, умноженное на $-a_{21}/a_{11}$. Затем, умножив первое уравнение на $-a_{31}/a_{11}$ и прибавив результат к третьему уравнению, также исключим из него x_1 . Получим равносильную (5) систему уравнений вида

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1, \\
a'_{22}x_2 + a'_{23}x_3 &= b'_2, \\
a'_{32}x_2 + a'_{33}x_3 &= b'_3; \\
a'_{ij} &= a_{ij} - \frac{a_{i1}}{a_{11}} a_{1j}, \quad i, j = 2, 3, \\
b'_i &= b_i - \frac{a_{i1}}{a_{11}} b_1, \quad i = 2, 3.
\end{aligned} \tag{6}$$

Теперь из третьего уравнения системы (6) нужно исключить x_2 . Для этого умножим второе уравнение на $-a'_{32}/a'_{22}$ и прибавим результат к третьему. Получим

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1, \\
a'_{22}x_2 + a'_{23}x_3 &= b'_2, \\
a''_{33}x_3 &= b''_3; \\
a''_{33} &= a'_{33} - \frac{a'_{32}}{a'_{22}} a'_{23}, \quad b''_3 = b'_3 - \frac{a'_{32}}{a'_{22}} b'_2.
\end{aligned} \tag{7}$$

Матрица системы (7) имеет треугольный вид. На этом заканчивается прямой ход метода Гаусса.

Заметим, что в процессе исключения неизвестных приходится выполнять операции деления на коэффициенты a_{11} , a'_{22} и т. д. Поэтому они должны быть отличны от нуля. В противном случае необходимо соответственным образом переставить уравнения системы. Перестановка уравнений должна быть предусмотрена в вычислительном алгоритме при его реализации на компьютере.

Обратный ход начинается с решения третьего уравнения системы (7):

$$x_3 = b''_3 / a''_{33}$$

Используя это значение, можно найти x_2 из второго уравнения, а затем x_1 из первого:

$$\begin{aligned}
x_2 &= \frac{1}{a'_{22}} (b'_2 - a'_{23}x_3), \\
x_1 &= \frac{1}{a_{11}} (b_1 - a_{12}x_2 - a_{13}x_3).
\end{aligned}$$

Аналогично строится вычислительный алгоритм для линейной системы с произвольным числом уравнений.

Одной из модификаций метода Гаусса является *схема с выбором главного элемента*. Она состоит в том, что требование неравенства нулю диагональных элементов a_{ii} , на которые происходит деление в процессе исключений, заменяется более жестким: из всех оставшихся в i -м столбце элементов нужно выбрать наибольший по модулю и переставить уравнения так, чтобы этот элемент оказался на месте элемента a_{ii} . Диагональные элементы матрицы называются *ведущими* элементами; ведущий элемент a_{ii} – это коэффициент при i -м неизвестном в i -м уравнении на i -м шаге исключения.

Благодаря выбору наибольшего по модулю ведущего элемента уменьшаются множители, используемые для преобразования уравнений, что способствует снижению погрешностей вычислений. Поэтому метод Гаусса с выбором главного элемента обеспечивает приемлемую точность решения для не слишком большого числа ($n \leq 1000$) уравнений. Только для плохо обусловленных систем решения, полученные по этому методу, ненадежны.

Метод Гаусса целесообразно использовать для решения систем с плотно заполненной матрицей. Все элементы матрицы и правые части системы уравнений находятся в оперативной памяти машины. Объем вычислений определяется порядком системы n : число арифметических операций примерно равно $(2/3)n^3$.

Погрешности решения систем линейных уравнений методом Гаусса. Запишем систему в матричном виде: $A\vec{x} = \vec{b}$. Решение этой системы можно представить в виде $\vec{x} = A^{-1}\vec{b}$. Однако вычисленное по методу Гаусса решение \vec{x}_* отличается от этого решения из-за погрешностей округлений, связанных с ограниченностью разрядной сетки машины.

Существуют две величины, характеризующие степень отклонения полученного решения от точного. Одна из них – *погрешность* $\Delta\vec{x}$, равная разности этих значений; другая – *невязка* \vec{r} , равная разности между левой и правой частями уравнений при подстановке в них решения:

$$\Delta\vec{x} = \vec{x} - \vec{x}_*, \quad \vec{r} = A\vec{x}_* - \vec{b}$$

Можно показать, что если одна из этих величин равна нулю, то и другая должна равняться

нулю. Однако из малости одной не следует малость другой. При $\Delta \vec{x} \approx 0$ обычно и $\vec{r} \approx 0$, но обратное утверждение справедливо не всегда. В частности, для плохо обусловленных систем при $\vec{r} \approx 0$ погрешность решения может быть большой.

Вместе с тем в практических расчетах, если система не является плохо обусловленной, контроль точности решения осуществляется с помощью невязки (погрешность же обычно вычислить невозможно, поскольку неизвестно точное решение). Можно отметить, что метод Гаусса с выбором главного элемента в этих случаях дает малые невязки.

Понятия погрешности и невязки используются при численном решении не только систем линейных уравнений, но и других задач. В зависимости от задачи погрешность и невязка могут быть величинами скалярными, векторными (как в данном случае), матричными и др.

Нахождение определителя и обратной матрицы. Непосредственное нахождение определителя требует большого объема вычислений. Вместе с тем легко вычисляется определитель треугольной матрицы: он равен произведению ее диагональных элементов.

Для приведения матрицы к треугольному виду может быть использован *метод исключения*, т. е. прямой ход метода Гаусса. В процессе исключения элементов величина определителя не меняется. Знак определителя меняется на противоположный при перестановке его столбцов или строк. Следовательно, значение определителя после приведения матрицы A к треугольному виду вычисляется по формуле

$$\det A = (-1)^k \prod_{i=1}^n a_{ii}.$$

Здесь диагональные элементы a_{ii} берутся из преобразованной (а не исходной) матрицы. Через k обозначено число перестановок строк (или столбцов) матрицы при ее приведении к треугольному виду (для получения ненулевого или максимального по модулю ведущего элемента на каждом этапе исключения). Благодаря методу исключения можно вычислять определители 1000-го и большего порядков, и объем вычислений значительно меньше, чем в проведенных ранее оценках.

Теперь найдем обратную матрицу A^{-1} . Обозначим ее элементы через z_{ij} . Запишем равенство $AA^{-1} = E$ в виде

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \cdot \begin{pmatrix} z_{11} & z_{12} & \dots & z_{1n} \\ z_{21} & z_{22} & \dots & z_{2n} \\ \dots & \dots & \dots & \dots \\ z_{n1} & z_{n2} & \dots & z_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

Отсюда следует, что

$$Az_j = e_j, \quad j = 1, 2, \dots, n, \quad (8)$$

где z_j и e_j – j -е столбцы матриц A^{-1} и E соответственно (в столбце e_j равны нулю все компоненты, кроме j -й, которая равна единице). Таким образом, для нахождения j -го столбца обратной матрицы нужно решить систему уравнений (8). Решив n таких систем для $j = 1, 2, \dots, n$, мы найдем все столбцы z_j и, следовательно, саму обратную матрицу.

Поскольку при разных j матрица A системы (8) не меняется, исключение неизвестных при использовании метода Гаусса (прямой ход) проводится только один раз, причем сразу для всех правых частей – столбцов e_j . Затем для каждой из систем (8) делается обратный ход с соответствующей преобразованной правой частью.

Это весьма экономичный способ обращения матрицы, он требует примерно лишь в три раза больше действий, чем при решении одной системы уравнений.

Степень отклонения вычисленной обратной матрицы A_*^{-1} от ее точного значения характеризуется погрешностью ΔA^{-1} и невязкой R :

$$\Delta A^{-1} = A^{-1} - A_*^{-1}, \quad R = AA_*^{-1} - E.$$

Другие прямые методы. Среди прямых методов наиболее распространен метод Гаусса; он удобен для вычислений на компьютере. Вот некоторые другие методы.

Схема Жордана при выборе главного элемента не учитывает коэффициенты тех уравнений, из которых уже выбирался главный элемент. Она не имеет преимуществ по сравнению с методом Гаусса. Отметим лишь, что здесь облегчается обратный ход, поскольку система приводится к диагональному виду (а не к треугольному). Эта схема часто используется для нахождения обратной матрицы.

Метод квадратного корня используется в тех случаях, когда матрица системы является

симметричной.

Метод оптимального исключения удобен при построчном вводе матрицы системы в оперативную память. Однако построчный ввод имеет и недостатки: частые обращения к внешним устройствам, невозможность выбора главного элемента и др.

Клеточные методы могут использоваться для решения больших систем, когда матрица и вектор правых частей целиком не помещаются в оперативной памяти.

Итерационные методы

Уточнение решения. Решения, получаемые с помощью прямых методов, обычно содержат погрешности, вызванные округлениями при выполнении операций над числами с плавающей точкой на компьютере с ограниченным числом разрядов. В ряде случаев эти погрешности могут быть значительными, и необходимо найти способ их уменьшения. Рассмотрим здесь один из методов, позволяющий уточнить решение, полученное с помощью прямого метода.

Найдем решение системы линейных уравнений

$$A\vec{x} = \vec{b} \quad (9)$$

Пусть с помощью некоторого прямого метода вычислено приближенное решение $\vec{x}^{(0)}$ (т. е. найдены приближенные значения неизвестных $x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}$), называемое *начальным* или *нулевым приближением* к решению. Подставляя это решение в левую часть системы (9), получаем некоторый столбец правых частей $\vec{b}^{(0)}$, отличный от \vec{b} :

$$A\vec{x}^{(0)} = \vec{b}^{(0)} \quad (10)$$

Введем обозначения: $\Delta\vec{x}^{(0)}$ – погрешность полученного решения, $\vec{r}^{(0)}$ – невязка, т. е.

$$\Delta\vec{x}^{(0)} = \vec{x} - \vec{x}^{(0)}, \quad \vec{r}^{(0)} = A\vec{x}^{(0)} - \vec{b} = \vec{b}^{(0)} - \vec{b} \quad (11)$$

Вычитая равенство (10) из равенства (9), с учетом обозначений (11) получаем

$$A\Delta\vec{x}^{(0)} = -\vec{r}^{(0)} \quad (12)$$

Решая эту систему, находим значение погрешности $\Delta\vec{x}^{(0)}$, которое используем в качестве поправки к приближенному решению $\vec{x}^{(0)}$, вычисляя таким образом новое приближенное решение $\vec{x}^{(1)}$ (или *следующее приближение* к решению):

$$\vec{x}^{(1)} = \vec{x}^{(0)} + \Delta\vec{x}^{(0)}$$

Таким же способом можно найти новую поправку к решению $\Delta\vec{x}^{(1)}$ и следующее приближение $\vec{x}^{(2)} = \vec{x}^{(1)} + \Delta\vec{x}^{(1)}$ и т. д. Процесс продолжается до тех пор, пока очередное значение погрешности (поправки) $\Delta\vec{x}^{(k)}$ не станет достаточно малым, т. е. пока очередные приближенные значения неизвестных $x_1^{(k+1)}, x_2^{(k+1)}, \dots, x_n^{(k+1)}$ не будут мало отличаться от предыдущих значений $x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}$.

Рассмотренный процесс уточнения решения представляет собой итерационный метод решения системы линейных уравнений. Для нахождения очередного приближения, т. е. на каждой итерации, решаются системы уравнений вида (12) с одной и той же матрицей, являющейся матрицей исходной системы (9), при разных правых частях. Это позволяет строить экономичные алгоритмы. Например, при использовании метода Гаусса сокращается объем вычислений на этапе прямого хода.

Решение систем линейных уравнений с помощью итерационного метода (а также решение итерационными методами уравнений другого вида и их систем) сводится к следующему алгоритму. Вводятся исходные данные (коэффициенты уравнений и допустимое значение погрешности).

Задаются начальные приближения значений неизвестных (вектор-столбец $\vec{x}^{(0)}$). Они либо вводятся в компьютер, либо вычисляются каким-либо способом (в частности, путем решения системы уравнений с помощью прямого метода). Затем организуется циклический вычислительный процесс, каждый цикл которого представляет собой одну итерацию – переход от предыдущего приближения $\vec{x}^{(k-1)}$ к последующему $\vec{x}^{(k)}$. Если оказывается, что с увеличением числа итераций приближенное решение стремится к точному:

$$\lim_{k \rightarrow \infty} \vec{x}^{(k)} = \vec{x}$$

то итерационный метод называют *сходящимся*.

На практике наличие сходимости и достижение требуемой точности обычно определяют приближенно, поступая следующим образом. При малом (с заданной допустимой погрешностью) изменении \vec{x} на двух последовательных итерациях, т. е. при малом отличии $\vec{x}^{(k)}$ от $\vec{x}^{(k-1)}$ процесс прекращается, и происходит вывод значений неизвестных, полученных на последней итерации.

Возможны разные подходы к определению малости отличия \vec{x} на двух последовательных итерациях. Например, если задана допустимая погрешность $\varepsilon > 0$, то критерием окончания итерационного процесса можно считать выполнение одного из трех неравенств:

$$|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}| = \sqrt{\sum_{i=1}^n (x_i^{(k)} - x_i^{(k-1)})^2} < \varepsilon, \quad (13)$$

$$\max_{1 \leq i \leq n} |x_i^{(k)} - x_i^{(k-1)}| < \varepsilon, \quad (14)$$

$$\max_{1 \leq i \leq n} \left| \frac{x_i^{(k)} - x_i^{(k-1)}}{x_i^{(k)}} \right| < \varepsilon, \quad \text{при } |x_i| \gg 1. \quad (15)$$

Здесь в первом случае отличие векторов $\vec{x}^{(k)}$ и $\vec{x}^{(k-1)}$ «на ε » понимается в смысле малости модуля их разности, во втором – в смысле малости разностей всех соответствующих компонент векторов, в третьем – в смысле малости относительных разностей компонент. Если система не является плохо обусловленной, то в качестве критерия окончания итерационного процесса можно использовать и условие малости невязки, например

$$|\mathbf{r}^{(k)}| < \varepsilon. \quad (16)$$

В случае отсутствия сходимости, для предотвращения непроизводительных затрат машинного времени в алгоритм вводят счетчик числа итераций и при достижении им некоторого заданного значения счет прекращают и выводят текущие значения неизвестных.

Метод простой итерации. Этот метод широко используется для численного решения уравнений и их систем различных видов. Рассмотрим применение метода простой итерации к решению систем линейных уравнений.

Запишем исходную систему уравнений в векторно-матричном виде (2) и выполним ряд тождественных преобразований:

$$\begin{aligned} \mathbf{Ax} &= \mathbf{b}; & \mathbf{0} &= \mathbf{b} - \mathbf{Ax}; & \mathbf{x} &= \mathbf{b} - \mathbf{Ax} + \mathbf{x}; \\ \mathbf{x} &= (\mathbf{b} - \mathbf{Ax})\tau + \mathbf{x}; & \mathbf{x} &= (\mathbf{E} - \tau\mathbf{A})\mathbf{x} + \tau\mathbf{b}; \\ & & \mathbf{x} &= \mathbf{Bx} + \tau\mathbf{b}, \end{aligned} \quad (17)$$

где $\tau \neq 0$ – некоторое число, \mathbf{E} – единичная матрица, $\mathbf{B} = \mathbf{E} - \tau\mathbf{A}$. Получившаяся система (17) эквивалентна исходной системе и служит основой для построения метода простой итерации.

Выберем некоторое начальное приближение $\vec{x}^{(0)}$ и подставим его в правую часть системы (17):

$$\mathbf{x}^{(1)} = \mathbf{Bx}^{(0)} + \tau\mathbf{b}.$$

Поскольку $\vec{x}^{(0)}$ не является решением системы, в левой части (17) получится некоторый столбец $\vec{x}^{(1)}$, в общем случае отличный от $\vec{x}^{(0)}$. Полученный столбец $\vec{x}^{(1)}$ будем следующим (первым) приближением к решению. Аналогично, по известному k -му приближению можно найти $(k + 1)$ -е приближение:

$$\mathbf{x}^{(k+1)} = \mathbf{Bx}^{(k)} + \tau\mathbf{b}, \quad k = 0, 1, 2, \dots \quad (18)$$

Формула (18) и выражает собой метод простой итерации. Для ее применения нужно задать некий параметр τ . От значения τ зависит, будет ли сходиться метод, а если будет, то какова будет *скорость сходимости*, т. е. как много итераций нужно совершить для достижения требуемой точности. В частности, справедлива следующая теорема.

Т е о р е м а. Пусть $\det \mathbf{A} \neq 0$. Метод простой итерации (18) сходится тогда и только тогда, когда все собственные числа матрицы $\mathbf{B} = \mathbf{A} - \tau\mathbf{E}$ по модулю меньше единицы.

Для некоторых типов матрицы A можно указать правило выбора τ , обеспечивающее сходимость метода и оптимальную скорость сходимости. В простейшем же случае τ можно положить равным некоторому постоянному числу, например, 1, 0,1 и т.д.

Метод Гаусса-Зейделя. Одним из самых распространенных итерационных методов, отличающийся простотой и легкостью программирования, является *метод Гаусса-Зейделя*.

Проиллюстрируем сначала этот метод на примере решения системы трех линейных уравнений

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3. \end{aligned} \quad (19)$$

Предположим, что диагональные элементы a_{11} , a_{22} , a_{33} отличны от нуля (в противном случае можно переставить уравнения). Выразим неизвестные x_1 , x_2 и x_3 соответственно из первого, второго и третьего уравнений системы (19):

$$x_1 = \frac{1}{a_{11}}(b_1 - a_{12}x_2 - a_{13}x_3) \quad (20)$$

$$x_2 = \frac{1}{a_{22}}(b_2 - a_{21}x_1 - a_{23}x_3) \quad (21)$$

$$x_3 = \frac{1}{a_{33}}(b_3 - a_{31}x_1 - a_{32}x_2) \quad (22)$$

Зададим некоторые начальные (нулевые) приближения значений неизвестных: $x_1 = x_1^{(0)}$, $x_2 = x_2^{(0)}$, $x_3 = x_3^{(0)}$. Подставляя эти значения в правую часть выражения (20), получаем новое (первое) приближение для x_1 :

$$x_1^{(1)} = \frac{1}{a_{11}}(b_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)}).$$

Используя это значение для x_1 и приближение $x_3^{(0)}$ для x_3 , находим из (21) первое приближение для x_2 :

$$x_2^{(1)} = \frac{1}{a_{22}}(b_2 - a_{21}x_1^{(1)} - a_{23}x_3^{(0)}).$$

И наконец, используя вычисленные значения $x_1 = x_1^{(1)}$, $x_2 = x_2^{(1)}$, находим с помощью выражения (22) первое приближение для x_3 :

$$x_3^{(1)} = \frac{1}{a_{33}}(b_3 - a_{31}x_1^{(1)} - a_{32}x_2^{(1)}).$$

На этом заканчивается первая итерация решения системы (20) – (22). Используя теперь значения $x_1^{(1)}$, $x_2^{(1)}$, $x_3^{(1)}$, можно таким же способом провести вторую итерацию, в результате которой будут найдены вторые приближения к решению: $x_1 = x_1^{(2)}$, $x_2 = x_2^{(2)}$, $x_3 = x_3^{(2)}$ и т. д.

Приближение с номером k можно вычислить, зная приближение с номером $k - 1$, как

$$\begin{aligned} x_1^{(k)} &= \frac{1}{a_{11}}(b_1 - a_{12}x_2^{(k-1)} - a_{13}x_3^{(k-1)}), \\ x_2^{(k)} &= \frac{1}{a_{22}}(b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k-1)}), \\ x_3^{(k)} &= \frac{1}{a_{33}}(b_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)}). \end{aligned}$$

Итерационный процесс продолжается до тех пор, пока значения $x_1^{(k)}$, $x_2^{(k)}$, $x_3^{(k)}$ не станут близкими с заданной погрешностью к значениям $x_1^{(k-1)}$, $x_2^{(k-1)}$, $x_3^{(k-1)}$.

П р и м е р. Решить с помощью метода Гаусса-Зейделя следующую систему уравнений:

$$\begin{aligned} 4x_1 - x_2 + x_3 &= 4, \\ 2x_1 + 6x_2 - x_3 &= 7, \\ x_1 + 2x_2 - 3x_3 &= 0. \end{aligned}$$

Легко проверить, что решение данной системы следующее: $x_1 = x_2 = x_3 = 1$.

Р е ш е н и е. Выразим неизвестные x_1 , x_2 , x_3 соответственно из первого, второго и третьего уравнений:

$$x_1 = \frac{1}{4}(4 + x_2 - x_3), \quad x_2 = \frac{1}{6}(7 - 2x_1 + x_3), \quad x_3 = \frac{1}{3}(x_1 + 2x_2).$$

В качестве начального приближения (как это обычно делается) примем $x_1 = 0$, $x_2 = 0$, $x_3 = 0$. Найдем новые приближения неизвестных:

$$x_1^{(1)} = \frac{1}{4}(4 + 0 - 0) = 1, \quad x_2^{(1)} = \frac{1}{6}(7 - 2 \cdot 1 + 0) = \frac{5}{6}$$

$$x_3^{(1)} = \frac{1}{3} \left(1 + 2 \cdot \frac{5}{6} \right) = \frac{8}{9}.$$

Аналогично вычислим следующие приближения:

$$x_1^{(2)} = \frac{1}{4} \left(4 + \frac{5}{6} - \frac{8}{9} \right) = \frac{71}{72}, \quad x_2^{(2)} = \frac{1}{6} \left(7 - 2 \cdot \frac{71}{72} + \frac{8}{9} \right) = \frac{71}{72},$$

$$x_3^{(2)} = \frac{1}{3} \left(1 + 2 \cdot \frac{71}{72} \right) = \frac{71}{36}.$$

Итерационный процесс можно продолжать до получения малой разности между значениями неизвестных в двух последовательных итерациях.

Рассмотрим теперь систему n линейных уравнений с n неизвестными. Запишем ее в виде

$$a_{i1}x_1 + \dots + a_{i,i-1}x_{i-1} + a_{ii}x_i + a_{i,i+1}x_{i+1} + \dots + a_{in}x_n = b_i, \quad i = 1, 2, \dots, n.$$

Здесь также будем, предполагать, что все диагональные элементы отличны от нуля. Тогда в соответствии с методом Гаусса-Зейделя k -е приближение к решению можно представить в виде

$$x_i^{(k)} = \frac{1}{a_{ii}} \left(b_i - a_{i1}x_1^{(k)} - \dots - a_{i,i-1}x_{i-1}^{(k)} - a_{i,i+1}x_{i+1}^{(k-1)} - \dots - a_{in}x_n^{(k-1)} \right), \quad i = 1, 2, \dots, n. \quad (23)$$

Итерационный процесс продолжается до тех пор, пока все значения $x_i^{(k)}$ не станут близкими к $x_i^{(k-1)}$, т. е. в качестве критерия завершения итераций используется одно из условий (13)–(15), (16).

Для сходимости итерационного процесса (23) достаточно, чтобы модули диагональных коэффициентов для каждого уравнения системы были не меньше сумм модулей всех остальных коэффициентов (преобладание диагональных элементов):

$$|a_{ij}| \geq \sum_{j \neq i} |a_{ij}|, \quad i = 1, 2, \dots, n. \quad (24)$$

При этом хотя бы для одного уравнения неравенство должно выполняться строго. Эти условия являются достаточными для сходимости метода, но они не являются необходимыми, т. е. для некоторых систем итерации сходятся и при нарушении условий (24).