

## Решение систем линейных алгебраических уравнений

Решить систему методом Гаусса и матричным методом.

$$1. \begin{cases} 4x_1 + x_2 + x_3 + 2x_4 = 2 \\ x_1 + 3x_2 + 2x_3 - x_4 = 2 \\ 2x_1 - x_2 + 5x_3 + 3x_4 = -1 \\ 4x_1 + 5x_2 + 4x_3 - 4x_4 = 8 \end{cases}$$

$$2. \begin{cases} x_1 - x_2 + 2x_3 - x_4 = 1 \\ 2x_1 + 3x_3 + x_4 = 4 \\ x_1 + x_2 + 3x_3 - x_4 = 2 \\ 2x_1 + x_2 + 5x_3 - 2x_4 = 3 \end{cases}$$

$$3. \begin{cases} x_1 + 2x_2 - x_3 - x_4 = 0 \\ 2x_1 + 3x_2 - x_3 + x_4 = 3 \\ 2x_1 + 5x_2 + 2x_3 + x_4 = 3 \\ 3x_1 + 5x_2 + x_3 + 2x_4 = 5 \end{cases}$$

$$4. \begin{cases} x_1 + x_2 + x_3 + x_4 = 2 \\ x_1 + 2x_2 + 2x_3 + 2x_4 = 4 \\ 2x_1 + 3x_2 + 4x_3 + 4x_4 = 7 \\ 3x_1 + 4x_2 + 5x_3 + 6x_4 = 9 \end{cases}$$

$$5. \begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1 + 2x_2 + 2x_3 + x_4 = 0 \\ 2x_1 + 3x_2 + 4x_3 + 2x_4 = 0 \\ 3x_1 + 4x_2 + 5x_3 + 3x_4 = 0 \end{cases}$$

$$6. \begin{cases} 2x_1 + x_2 + x_3 + x_4 = 2 \\ 2x_1 + 2x_2 + x_3 + x_4 = 3 \\ 2x_1 + 2x_2 + 2x_3 + x_4 = 3 \\ 2x_1 + 2x_2 + 2x_3 + 3x_4 = 1 \end{cases}$$

$$7. \begin{cases} 2x_1 + x_2 + x_3 + x_4 = 1 \\ 2x_1 + 2x_2 + 2x_3 + 3x_4 = -1 \\ 4x_1 + 3x_2 + 3x_3 + 3x_4 = 1 \\ 6x_1 + 4x_2 + 5x_3 + 2x_4 = 4 \end{cases}$$

$$8. \begin{cases} x_1 + 2x_2 + 2x_3 + x_4 = 0 \\ 2x_1 + 3x_2 + 3x_3 + 2x_4 = 0 \\ 3x_1 + 5x_2 + 4x_3 + 2x_4 = 1 \\ 4x_1 + 7x_2 + 6x_3 + 2x_4 = 2 \end{cases}$$

$$9. \begin{cases} 2x_1 + x_2 + x_3 + x_4 = 2 \\ 2x_1 + 2x_2 + 2x_3 + 3x_4 = 1 \\ 2x_1 + 2x_2 + 3x_3 + 4x_4 = 0 \\ 2x_1 + 2x_2 + 3x_3 + 5x_4 = -1 \end{cases}$$

$$10. \begin{cases} 2x_1 + 2x_2 + x_3 + x_4 = 1 \\ 2x_1 + 3x_2 + 2x_3 + x_4 = 1 \\ 4x_1 + 5x_2 + 4x_3 + 3x_4 = 1 \\ 6x_1 + 5x_2 + x_3 + x_4 = 5 \end{cases}$$

$$11. \begin{cases} 2x_1 + 3x_2 + 3x_3 + 3x_4 = 5 \\ 2x_1 + 2x_2 + 2x_3 + 3x_4 = 5 \\ 2x_1 + 2x_2 + x_3 + 2x_4 = 4 \\ 2x_1 + 2x_2 + x_3 + x_4 = 3 \end{cases}$$

$$12. \begin{cases} x_1 + 2x_2 + x_3 + 2x_4 = 1 \\ x_1 + 4x_2 + 2x_3 + x_4 = 2 \\ 2x_1 + 6x_2 + x_3 + 3x_4 = 5 \\ 2x_1 + 5x_2 + 2x_3 + 2x_4 = 3 \end{cases}$$

$$13. \begin{cases} x_1 + 2x_2 - x_4 = 0 \\ 2x_1 + x_2 + 3x_3 + x_4 = 3 \\ x_1 + 2x_2 + 3x_3 - x_4 = 0 \\ 2x_1 + 2x_2 + 5x_3 - x_4 = 1 \end{cases}$$

$$14. \begin{cases} x_1 + 2x_2 + x_3 + 2x_4 = 3 \\ 2x_1 + 3x_2 - x_3 + 2x_4 = 5 \\ 2x_1 + 5x_2 + 2x_3 + 2x_4 = 7 \\ 3x_1 + 5x_2 + x_3 + 4x_4 = 8 \end{cases}$$

$$15. \begin{cases} x_1 + x_2 + 2x_3 + x_4 = 0 \\ x_1 + 2x_2 + 4x_3 + 2x_4 = 1 \\ 2x_1 + 3x_2 + 8x_3 + 4x_4 = 2 \\ 3x_1 + 4x_2 + 10x_3 + 6x_4 = 3 \end{cases}$$

$$16. \begin{cases} x_1 + x_2 + x_3 = 0 \\ x_1 + 2x_2 + 2x_3 + 2x_4 = 1 \\ 2x_1 + 3x_2 + 4x_3 + 4x_4 = 1 \\ 3x_1 + 4x_2 + 5x_3 + 6x_4 = 1 \end{cases}$$

$$17. \begin{cases} x_1 + 2x_2 + 3x_3 + 3x_4 = 2 \\ x_1 + 3x_2 + 4x_3 + 3x_4 = 2 \\ 2x_1 + 3x_2 + 4x_3 + 5x_4 = 3 \\ 3x_1 + 5x_2 + 7x_3 + 7x_4 = 4 \end{cases}$$

$$18. \begin{cases} 2x_1 + x_2 + x_3 + 2x_4 = 2 \\ 2x_1 + 2x_2 + x_3 + 2x_4 = 3 \\ x_1 + x_2 + x_3 + x_4 = 2 \\ x_1 + x_2 + x_3 + 3x_4 = 2 \end{cases}$$

$$19. \begin{cases} x_1 + 2x_2 + 2x_3 + 2x_4 = 1 \\ 2x_1 + 3x_2 + 3x_3 + 4x_4 = 2 \\ 3x_1 + 5x_2 + 4x_3 + 4x_4 = 2 \\ 4x_1 + 7x_2 + 6x_3 + 4x_4 = 0 \end{cases}$$

$$20. \begin{cases} 3x_1 + 2x_2 + x_3 + x_4 = 1 \\ 3x_1 + 3x_2 + 2x_3 + 2x_4 = -1 \\ 3x_1 + 3x_2 + 3x_3 + 4x_4 = -4 \\ 3x_1 + 3x_2 + 3x_3 + 5x_4 = -5 \end{cases}$$

$$21. \begin{cases} 5x_1 + 3x_2 + 3x_3 + 3x_4 = 8 \\ 2x_1 - 2x_2 - 3x_3 + 3x_4 = 5 \\ 2x_1 + 2x_2 - 3x_3 + 2x_4 = 4 \\ 2x_1 + 2x_2 + x_3 + 2x_4 = 4 \end{cases}$$

$$22. \begin{cases} 4x_1 + 2x_2 + x_3 + 2x_4 = 6 \\ x_1 + 4x_2 + 2x_3 + x_4 = 5 \\ 2x_1 + 6x_2 + x_3 + 3x_4 = 8 \\ 2x_1 + 5x_2 + 2x_3 + 2x_4 = 7 \end{cases}$$

$$23. \begin{cases} 3x_1 + 2x_2 - x_4 = 2 \\ 2x_1 + x_2 + 3x_3 - x_4 = 1 \\ x_1 + 2x_2 + 5x_3 - x_4 = 0 \\ 2x_1 + 2x_2 + 5x_3 - x_4 = 1 \end{cases}$$

$$24. \begin{cases} x_1 + 2x_2 + x_3 + 2x_4 = 3 \\ 2x_1 + 3x_2 - x_3 + 2x_4 = 5 \\ 2x_1 + 5x_2 + 2x_3 + 2x_4 = 7 \\ 3x_1 + 5x_2 + x_3 + 4x_4 = 8 \end{cases}$$

$$25. \begin{cases} x_1 + 3x_2 + 2x_3 + x_4 = 0 \\ x_1 + 4x_2 + 4x_3 + 2x_4 = 1 \\ 2x_1 + 3x_2 + 8x_3 + 4x_4 = 2 \\ 3x_1 + 4x_2 + 10x_3 + 6x_4 = 3 \end{cases}$$

$$26. \begin{cases} x_1 + x_2 + x_3 = 0 \\ x_1 + 2x_2 + 2x_3 + 2x_4 = 1 \\ 5x_1 + 3x_2 + 4x_3 + 4x_4 = -1 \\ 3x_1 + 4x_2 + 5x_3 + 6x_4 = 1 \end{cases}$$

$$27. \begin{cases} x_1 - 2x_2 + 3x_3 + 3x_4 = 2 \\ x_1 + 3x_2 - 4x_3 + 3x_4 = 2 \\ 2x_1 + 3x_2 - 4x_3 + 5x_4 = 3 \\ 3x_1 - 5x_2 + 7x_3 + 7x_4 = 4 \end{cases}$$

$$28. \begin{cases} 5x_1 + x_2 + x_3 + 2x_4 = 2 \\ 2x_1 + 4x_2 + x_3 + 2x_4 = 5 \\ x_1 + x_2 + 3x_3 + x_4 = 4 \\ x_1 + x_2 - x_3 + 3x_4 = 0 \end{cases}$$

## ГЛАВА V

### СИСТЕМЫ УРАВНЕНИЙ

В главе V рассмотрены методы решения систем алгебраических уравнений. В § 1 изложено решение линейных систем методом исключения Гаусса, а также вычисление определителя и обращение матрицы; дан обзор других методов решения этих задач. В § 2 приведены различные методы нахождения корня одного трансцендентного уравнения. В § 3 некоторые из этих методов обобщены на системы нелинейных уравнений.

#### § 1. Линейные системы

**1. Задачи линейной алгебры.** Выделяют четыре основные задачи линейной алгебры: решение системы линейных уравнений  $Ax = b$ , где  $A$  — квадратная матрица и  $x, b$  — векторы; вычисление определителя; нахождение обратной матрицы; определение собственных значений и собственных векторов матрицы. В этом параграфе мы подробно рассмотрим первую задачу и попутно решим вторую и третью. Четвертая задача существенно сложнее, и ей посвящена следующая глава.

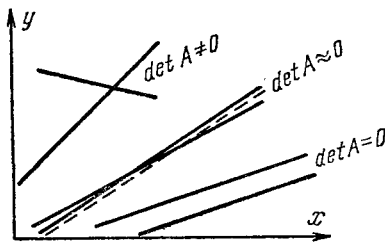


Рис. 24.

Известно, что если  $\det A = 0$ , то система линейных уравнений или не имеет решения, или имеет бесчисленное множество решений.

Если же  $\det A \neq 0$ , то система имеет решение, притом единственное. Далее мы будем рассматривать только последний случай.

Все эти случаи хорошо иллюстрируются геометрически на системе двух уравнений (рис. 24). Каждому уравнению соответствует прямая в плоскости  $x, y$ , а точка пересечения этих прямых есть решение системы (для  $n$  уравнений решение есть точка пересечения всех  $n$  гиперплоскостей в  $n$ -мерном пространстве). Если  $\det A = 0$ , то наклоны прямых равны, и они либо параллельны, либо совпадают. В противном случае прямые имеют единственную точку пересечения.

На практике кроме существования и единственности решения важна еще устойчивость относительно погрешностей правой части и элементов матрицы. Формально перепишем линейную систему в виде  $\mathbf{x} = A^{-1}\mathbf{b}$ . Варьируя это равенство и определяя вариацию обратной матрицы из соотношения  $\delta E = \delta(AA^{-1}) = A\delta A^{-1} + \delta A A^{-1} = 0$ , получим

$$\delta \mathbf{x} = A^{-1}(\delta \mathbf{b} - \delta A \cdot \mathbf{x}).$$

Формально устойчивость есть, ибо при  $\det A \neq 0$  обратная матрица существует. Но если матрица  $A^{-1}$  имеет большие элементы, то можно указать такой вид погрешности исходных данных, который сильно изменит решение. В этом случае систему называют плохо обусловленной (по-видимому, плохая обусловленность была известна еще Гауссу). Очевидно, у плохо обусловленных систем  $\det A \approx 0$ ; однако заметим, что этот признак плохой обусловленности является необходимым, но недостаточным.

Плохо обусловленная система геометрически соответствует почти параллельным прямым. При этом небольшое изменение наклона или сдвиг одной прямой сильно меняют положение точки пересечения (рис. 24, пунктир). В многомерном случае геометрическая картина может быть более сложной. Так, для трех переменных возможен случай плохой обусловленности, когда соответствующие трем уравнениям плоскости пересекаются под большими углами (т. е. далеки от параллельности), но линии их попарного пересечения почти параллельны.

В теоретических исследованиях обусловленность часто характеризуют числом  $\kappa = \|A\| \cdot \|A^{-1}\|$ . Это число зависит от того, какая норма матриц выбрана, но при любой норме  $\kappa \geq 1$ . Чем больше это число, тем хуже обусловленность системы; обычно  $\kappa \sim 10^3 - 10^4$  уже означает плохую обусловленность.

В практических расчетах этим определением плохой обусловленности пользуются редко, ибо для его проверки надо находить обратную матрицу, что при плохо обусловленной матрице  $A$  нелегко сделать. Чаще ограничиваются проверкой условия  $\det A \approx 0$ , хотя оно является необходимым, но недостаточным, что видно из простого примера. Положим  $A = \varepsilon E$ , где  $E$  — единичная матрица; тогда  $\det A = \varepsilon^n$ , и даже при не очень малых  $\varepsilon$  детерминант высокого порядка  $n$  очень мал. Но система с диагональной матрицей хорошо обусловлена, и для нее критерий  $\kappa = \|A\| \cdot \|A^{-1}\| = 1$  наиболее благоприятен.

Методы решения линейных систем делятся на прямые и итерационные. Прямые методы дают решение за конечное число действий, просты и наиболее универсальны; они рассматриваются в этом параграфе. Для систем небольшого порядка  $n \leq 200$  применяются практически только прямые методы. Итерационные методы приведены в § 3; они выгодны для систем специального вида, со слабо заполненной матрицей очень большого порядка  $n \approx 10^3 - 10^5$ . Сравнительно недавно для решения плохо обусловленных систем стали применять методы регуляризации.



уравнения с ненулевыми элементами ниже главной диагонали:

$$\sum_{j=k}^n a_{ij}^{(k)} x_j = b_i^{(k)}, \quad k \leq i \leq n. \quad (3)$$

Умножим  $k$ -ю строку на число

$$c_{mk} = a_{mk}^{(k)} / a_{kk}^{(k)}, \quad m > k, \quad (4)$$

и вычтем из  $m$ -й строки. Первый ненулевой элемент этой строки обратится в нуль, а остальные изменятся по формулам

$$\begin{aligned} a_{ml}^{(k+1)} &= a_{ml}^{(k)} - c_{mk} a_{kl}^{(k)}, \\ b_m^{(k+1)} &= b_m^{(k)} - c_{mk} b_k^{(k)}, \quad k < m, \quad l \leq n. \end{aligned} \quad (5)$$

Производя вычисления по этим формулам при всех указанных индексах, исключим элементы  $k$ -го столбца. Будем называть такое исключение *циклом* процесса. Выполнение всех циклов называется *прямым ходом* исключения.

Запишем треугольную систему, получающуюся после выполнения всех циклов. При приведении системы к треугольному виду освободятся клетки в нижней половине матрицы системы (1). На освободившиеся места матрицы поставим множители  $c_{mk}$ ; их следует запоминать, ибо они потребуются при обращении матрицы или уточнении решения. Получим

$$\begin{aligned} \sum_{k=i}^n a_{ik}^{(i)} x_k &= b_i^{(i)}, \quad 1 \leq i \leq n, \\ \begin{array}{cccccc} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \dots & a_{1n}^{(1)} & b_1^{(1)} \\ c_{21} & a_{22}^{(2)} & a_{23}^{(2)} & \dots & a_{2n}^{(2)} & b_2^{(2)} \\ c_{31} & c_{32} & a_{33}^{(3)} & \dots & a_{3n}^{(3)} & b_3^{(3)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{n1} & c_{n2} & c_{n3} & \dots & a_{nn}^{(n)} & b_n^{(n)} \end{array} \end{aligned} \quad (6)$$

Треугольная система (6) легко решается *обратным ходом* по формулам (2), в которых всем коэффициентам надо приписать сверху (в скобках) индекс строки.

Сделаем несколько замечаний. Исключение по формулам (4)–(5) нельзя проводить, если в ходе расчета на главной диагонали оказался нулевой элемент  $a_{kk}^{(k)} = 0$ . Но в первом столбце промежуточной системы (3) все элементы не могут быть нулями: это означало бы, что  $\det A = 0$ . Перестановкой строк можно переместить ненулевой элемент на главную диагональ и продолжить расчет.

Если элемент на главной диагонали  $a_{kk}^{(k)}$  мал, то эта строка умножается на большие числа  $c_{mk}$ , что приводит к значительным ошибкам округления при вычитаниях. Чтобы избежать этого, каждый цикл всегда начинают с перестановки строк. Среди элементов столбца  $a_{mk}^{(k)}$ ,  $m \geq k$ , находят *главный*, т. е. наибольший по модулю в  $k$ -м столбце, и перестановкой строк переводят его на главную диагональ, после чего делают исключения. В методе Гаусса с выбором главного элемента погрешность округления обычно невелика. Только для плохо обусловленных систем устойчивость этого метода оказывается недостаточной.

Погрешность округления можно еще уменьшить, если выбирать в каждом цикле элемент  $a_{ml}^{(k)}$ ,  $m, l \geq k$ , максимальный по модулю во всей матрице. Однако точность при этом возрастает не сильно по сравнению со случаем выбора главного элемента, а расчет заметно усложняется, ибо требуется перестановка не только строк, но и столбцов. Этот способ невыгоден для ЭВМ и применяется только при расчетах с небольшим количеством знаков на клавишных машинах.

Для контроля расчета полезно найти *невязки*:

$$r_k = b_k - \sum_{i=1}^n a_{ki}x_i, \quad 1 \leq k \leq n. \quad (7)$$

Если они велики, то это означает грубую ошибку в расчете (ошибка в программе, сбой ЭВМ). Если они малы, а система хорошо обусловлена, то решение найдено достаточно аккуратно. Правда, для плохо обусловленных систем малость невязок не гарантирует хорошей точности решения.

Метод Гаусса с выбором главного элемента надежен, прост и наиболее выгоден для линейных систем общего вида с плотно заполненной матрицей. Он требует примерно  $n^2$  ячеек в оперативной памяти ЭВМ, так что на БЭСМ-4 можно решать системы до 60 порядка. При вычислениях производится  $\sim 2/3 n^3$  арифметических действий; из них половина сложений, половина умножений и  $n$  делений.

3. **Определитель и обратная матрица** легко вычисляются методом исключения. В самом деле, вычитание строки из строки не меняет значение определителя. Значит, в процессе исключения элементов (4)—(5) абсолютная величина определителя не меняется, а знак может измениться благодаря перестановке строк. Определитель же треугольной матрицы (6) равен произведению диагональных элементов. Поэтому он вычисляется по формуле:

$$\det A = \pm \prod_{k=1}^n a_{kk}^{(k)}, \quad (8)$$

где знак зависит от того, четной или нечетной была суммарная перестановка строк. Для вычисления определителя требуется примерно  $n^2$  ячеек памяти и  $\frac{2}{3}n^3$  арифметических действий.

На примере вычисления определителя можно убедиться в экономичности хороших численных методов. Вспомним формальное определение определителя как суммы всевозможных произведений элементов, взятых из разных строк и столбцов. Таких произведений имеется  $n! \approx \sqrt{2\pi n} (n/e)^n$ , и прямое их вычисление уже при небольших  $n \approx 30$  требует астрономического числа действий — более  $10^{30}$ , что вряд ли когда-нибудь станет под силу ЭВМ. А метод исключения легко позволяет вычислять определители сотого и более порядка.

Перейдем к вычислению обратной матрицы. Обозначим ее элементы через  $\alpha_{lm}$ . Тогда соотношение  $AA^{-1} = E$  можно записать так:

$$\sum_{k=1}^n a_{ik} \alpha_{kl} = \delta_{il}, \quad 1 \leq i, l \leq n. \quad (9)$$

Видно, что если рассматривать  $l$ -й столбец обратной матрицы как вектор, то он является решением линейной системы (9) с матрицей  $A$  и специальной правой частью (в которой на  $l$ -м месте стоит единица, а на остальных — нули).

Таким образом, для обращения матрицы надо решить  $n$  систем линейных уравнений с одинаковой матрицей  $A$  и разными правыми частями. Приведение матрицы  $A$  к треугольной по формулам (4)—(5) делается при этом только один раз. В дальнейшем при помощи чисел  $c_{mk}$  по формуле (5) преобразуются все правые части, и для каждой правой части делается обратный ход.

При хорошей организации вычислений для обращения матрицы этим методом требуется примерно  $2n^2$  ячеек оперативной памяти ЭВМ и  $2n^3$  арифметических действий (можно уложиться в  $\frac{3}{2}n^2$  ячеек, но это сильно усложняет программу и увеличивает время счета). Заметим, что при обращении матриц контролировать расчет вычислением невязки  $R = E - AA^{-1}$  невыгодно: перемножение матриц требует столько же действий ( $2n^3$ ), как и обращение матрицы!

Любопытно отметить, что обращение матрицы сводится к решению  $n$  систем линейных уравнений, а требует лишь втрое больше действий, чем решение одной системы уравнений. Это объясняется тем, что при решении линейной системы большая часть вычислений связана с приведением матрицы к треугольному виду, что при обращении матрицы делается только один раз. Обратный ход и преобразования правых частей выполняются много быстрее.

Поэтому, если требуется несколько раз решить линейную систему с одной и той же матрицей, то выгодно привести матрицу к треугольной форме (6) только однажды, используя величины  $c_{mk}$  во всех последующих вычислениях.



**4. О других прямых методах.** Есть очень много других прямых методов решения задач линейной алгебры. Рассмотрим формальные характеристики наиболее известных методов.

Метод оптимального исключения имеет ту же скорость и требует той же памяти, что и метод Гаусса. Но если матрица вводится в оперативную память ЭВМ не вся сразу, а построчно, то для метода Гаусса требуется  $1/2n^2$  ячеек, а для метода оптимального исключения достаточно  $1/4n^2$  ячеек. Практическая ценность этого преимущества невелика, ибо построчный ввод означает много обращений к внешней памяти, т. е. сильное увеличение времени расчета. Кроме того, при построчном вводе невозможно выбрать главный элемент, что сказывается на ошибках округления.

Метод окаймления мало отличается от метода оптимального исключения и имеет те же характеристики.

Метод отражений требует вдвое большего числа действий, чем метод Гаусса (оперативная память та же).

Метод ортогонализации втрое медленнее метода Гаусса. Им интересовались в надежде на то, что он позволит решать плохо обусловленные системы. Но выяснилось, что при больших  $n$  сама ортогонализация приводит к большой потере точности (сравните с разложением функции по неортогональным функциям), и лучше использовать методы регуляризации.

Метод Жордана имеет ту же скорость, что и метод Гаусса; при решении линейных систем он не дает никаких преимуществ. Но при обращении матрицы он требует меньшей оперативной памяти — всего  $n^2$  ячеек.

Для решения хорошо обусловленных линейных систем общего вида метод Гаусса является одним из лучших; при обращении матрицы немного выгоднее метод Жордана. Но для систем специального вида (например, содержащих много нулевых элементов) существуют более быстрые методы. Некоторые из них будут изложены далее.

**5. Прогонка.** Пусть матрица  $A$  содержит много нулевых элементов, расположенных в матрице не беспорядочно, а плотными массивами на заранее известных местах. Тогда расчет по методу Гаусса можно организовать так, чтобы не включать эти элементы. Тем самым объем вычислений и требуемая память уменьшаются, зачастую очень сильно.

На рис. 25 приведены структуры матриц, которые нередко встречаются в задачах физики и техники и допускают такое ускорение расчета; горизонтальными линиями изображены положения ненулевых элементов, окаймлены границы массивов нулевых и ненулевых элементов. К таким матрицам относятся ленточные ( $a$ ), ящичные ( $b$ ), квазитреугольные ( $d$ ), почти треугольные ( $e$ ) и многие другие\*). Можно показать, что при обходе нулевых элементов решение системы с почти треугольной матрицей требует

---

\*) Напомним принятую терминологию. Матрица называется верхней треугольной, если все элементы ниже главной диагонали равны нулю ( $a_{ik}=0$  при  $i > k$ ); аналогично определяется нижняя треугольная матрица. Почти треугольной называется матрица, элементы которой удовлетворяют соотношению  $a_{ik}=0$  при  $i > k+1$ , т. е. ненулевые элементы имеются не только в верхнем треугольнике, но и в примыкающей к нему «боковой диагонали». Трехдиагональной называется матрица, у которой ненулевые элементы имеются только на главной диагонали и примыкающих к ней, т. е.  $a_{ik}=0$  при  $|i-k| > 1$ . Нетрудно записать определения других типов матриц, изображенных на рис. 25.

всего  $2n^2$  действий, а с ленточной — даже  $\frac{1}{2}k^2n$ , где  $k$  — ширина ленты, т. е. выигрыш во времени счета очень велик.

Выбор наибольшего элемента в таких расчетах делать нельзя, ибо перестановка столбцов разрушает специальную структуру матрицы. В матрицах с симметричной структурой недопустим

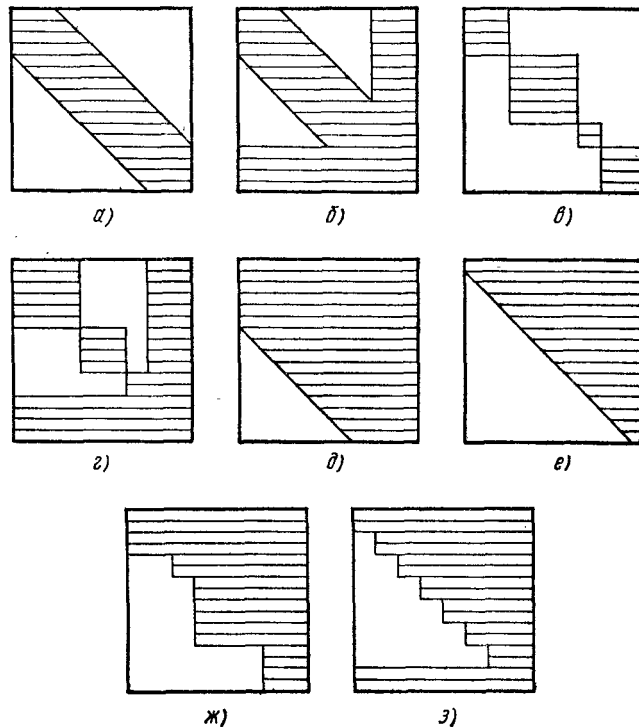


Рис. 25.

даже выбор главного элемента. Но обычно в этом нет необходимости, поскольку подобные физические задачи приводят, как правило, к хорошо обусловленным матрицам с большими элементами на главной диагонали, для которых ошибки округления в методе Гаусса невелики.

Наиболее важным частным случаем метода Гаусса является *метод прогонки*, применяемый к системам с трехдиагональной матрицей (они часто встречаются при решениях краевых задач для дифференциальных уравнений второго порядка). Такие системы обычно записывают в каноническом виде

$$\begin{aligned} a_i x_{i-1} - b_i x_i + c_i x_{i+1} &= d_i, & 1 \leq i \leq n, \\ a_1 &= c_n = 0. \end{aligned} \quad (10)$$

Формула (10) называется разностным уравнением второго порядка, или трехточечным уравнением. В этом случае прямой ход (без выбора главного элемента) сводится к исключению элементов  $a_i$ . Получается треугольная система, содержащая в каждом уравнении только два неизвестных,  $x_i$  и  $x_{i+1}$ . Поэтому формулы обратного хода имеют следующий вид:

$$x_i = \xi_{i+1}x_{i+1} + \eta_{i+1}, \quad i = n, n-1, \dots, 1. \quad (11)$$

Уменьшим в формуле (11) индекс на единицу и подставим в уравнение (10):

$$a_i (\xi_i x_i + \eta_i) - b_i x_i + c_i x_{i+1} = d_i.$$

Выражая отсюда  $x_i$  через  $x_{i+1}$ , получим

$$x_i = \frac{c_i}{b_i - a_i \xi_i} x_{i+1} + \frac{a_i \eta_i - d_i}{b_i - a_i \xi_i}.$$

Чтобы это выражение совпало с (11), надо, чтобы стоящие в его правой части дроби были равны соответственно  $\xi_{i+1}$  и  $\eta_{i+1}$ . Отсюда получим удобную запись формул прямого хода

$$\begin{aligned} \xi_{i+1} &= c_i / (b_i - a_i \xi_i), \\ \eta_{i+1} &= (a_i \eta_i - d_i) / (b_i - a_i \xi_i), \quad i = 1, 2, \dots, n. \end{aligned} \quad (12)$$

Попутно можно найти определитель трехдиагональной матрицы

$$\det A = \prod_{i=1}^n (a_i \xi_i - b_i). \quad (13)$$

Вычисления по формулам прогонки (12) — (11) требуют всего  $3n$  ячеек памяти и  $9n$  арифметических действий, т. е. они гораздо экономнее общих формул метода исключения.

В формулах прямого и обратного хода начало счета «замаскировано»: для начала (*развязки*) расчета формально требуется задать величины  $\xi_1$ ,  $\eta_1$  и  $x_{n+1}$ , которые неизвестны. Однако перед этими величинами в формулах стоят множители  $a_1$  или  $\xi_{n+1} \sim c_n$ , равные нулю. Это позволяет начать вычисления, полагая, например,  $\xi_1 = \eta_1 = x_{n+1} = 0$ .

Покажем, что если выполнено условие *преобладания диагональных элементов*

$$|b_i| \geq |a_i| + |c_i| \quad (14)$$

(причем хотя бы для одного  $i$  имеет место неравенство), то в формулах прямого хода (12) не возникает деления на нуль, и тем самым исходная система (10) имеет единственное решение. Для этого предположим, что  $|\xi_i| < 1$  при некотором значении индекса. Тогда легко проверяется цепочка неравенств

$$\begin{aligned} |\xi_{i+1}| &= |c_i| / |b_i - a_i \xi_i| \leq |c_i| / (|b_i| - |a_i| |\xi_i|) \leq \\ &\leq |c_i| / (|c_i| + |a_i| - |a_i| \times |\xi_i|) < 1. \end{aligned}$$

Поскольку можно положить  $\xi_1 = 0$ , отсюда по индукции следует  $|\xi_i| < 1$ ; значит,  $|b_i - a_i \xi_i| > |c_i| \geq 0$ , что и требовалось доказать. При выполнении условия (14) формулы прогонки не только безавостны, но и устойчивы относительно ошибок округления и позволяют успешно решать системы уравнений с несколькими сотнями неизвестных.

Условие (14) является достаточным, но не необходимым условием устойчивости прогонки. Конечно, можно построить примеры неустойчивости при несоблюдении этого условия. Но в практических расчетах для хорошо обусловленных систем типа (10) прогонка часто оказывается достаточно устойчивой даже при нарушении условия преобладания диагональных элементов.

Заметим, что к линейным системам с трехдиагональной матрицей обычно приводят трехточечные разностные схемы для дифференциальных уравнений второго порядка (глава VIII, § 2).

**6. Метод квадратного корня.** Этот метод пригоден только для линейных систем с эрмитовой \*) матрицей  $A = A^H$ , и формулы расчета при этом несколько сложнее, чем в методе Гаусса. Зато метод квадратного корня вдвое быстрее метода Гаусса.

Метод основан на представлении эрмитовой матрицы системы в виде произведения трех матриц

$$A = S^H D S. \quad (15)$$

Здесь  $D$  — диагональная матрица с элементами  $d_{ii} = \pm 1$ ,  $S$  — верхняя треугольная матрица ( $s_{ik} = 0$  при  $i > k$ ), а  $S^H$  — эрмитово сопряженная к ней нижняя треугольная матрица. Для полной определенности разложения потребуем вещественности и положительности диагональных элементов  $s_{ii} > 0$ .

Перепишем соотношение (15) в следующем виде:

$$a_{kl} = \sum_{i=1}^n s_{ik}^* d_{ii} s_{il} = \sum_{i=1}^{\min(k, l)} d_{ii} s_{ik}^* s_{il};$$

ограничение верхнего предела в сумме связано с обращением в нуль элементов  $S$  ниже главной диагонали. Последнее равенство можно записать в такой форме:

$$a_{kk} = \sum_{i=1}^k d_{ii} |s_{ik}|^2, \\ a_{kl} = \sum_{i=1}^k d_{ii} s_{ik}^* s_{il}, \quad k < l,$$

---

\*) Напомним, что матрица называется эрмитовой, если  $a_{ik} = a_{ki}^*$ ; эрмитова матрица с вещественными элементами является симметричной.

или окончательно

$$\begin{aligned} d_{kk} &= \text{sign} \left( a_{kk} - \sum_{i=1}^{k-1} d_{ii} |s_{ik}|^2 \right), \\ s_{kk} &= \sqrt{\left| a_{kk} - \sum_{i=1}^{k-1} d_{ii} |s_{ik}|^2 \right|}, \\ s_{kl} &= (a_{kl} - \sum_{i=1}^{k-1} d_{ii} s_{ik}^* s_{il}) / (s_{kk} d_{kk}) \quad \text{при } k+1 \leq l \leq n. \end{aligned} \quad (16)$$

В этих формулах сначала полагаем  $k=1$  и последовательно вычисляем все элементы первой строки матрицы  $S$ ; при  $k=1$  все суммы в формулах (16) отсутствуют. Затем полагаем  $k=2$  и вычисляем вторую строку и т. д.

Когда все элементы матриц найдены, то решение линейной системы  $Ax=b$  сводится к последовательному решению трех систем, двух треугольных и одной диагональной:

$$S^H z = b, \quad Dy = z, \quad Sx = y, \quad (17)$$

что делается обычным обратным ходом по формулам

$$\begin{aligned} y_1 &= b_1 / s_{11} d_{11}, \\ y_i &= (b_i - \sum_{l=1}^{i-1} d_{ll} y_l s_{il}^*) / s_{ii} d_{ii}, \quad i = 2, 3, \dots, n; \\ x_n &= y_n / s_{nn}, \\ x_i &= (y_i - \sum_{l=i+1}^n s_{il} x_l) / s_{ii}, \quad i = n-1, n-2, \dots, 1. \end{aligned} \quad (18)$$

Определитель матрицы вычисляется по формуле

$$\det A = \prod_{i=1}^n d_{ii} s_{ii}^2. \quad (19)$$

Метод квадратного корня требует примерно  $1/3 n^3$  арифметических действий, т. е. при больших  $n$  он вдвое быстрее метода Гаусса, и занимает вдвое меньше ячеек памяти. Это понятно, ибо метод использует информацию о симметрии матрицы.

Кроме того, для ленточной, ящичной и некоторых других структур матрицы  $A$  (рис. 25, а—г) матрица  $S$  будет иметь аналогичную структуру, т. е. содержать массивы нулевых элементов на заранее известных местах. Учет этого позволяет сильно сократить объем вычислений; как и в методе Гаусса. Однако заметим, что для ленточных матриц с узкой лентой, особенно для трехдиагональных, метод квадратного корня по скорости мало отличается от метода Гаусса и может быть даже медлен-

ней, ибо среди производящихся в нем действий есть извлечение корня, медленно выполняемое на ЭВМ.

Наиболее частый на практике случай эрмитовой матрицы — это вещественная симметричная матрица  $A$ . Тогда никаких комплексных чисел при вычислениях не возникает, так что матрица  $S$  тоже вещественная. Если вдобавок матрица  $A$  положительно определенная (для этого необходима и достаточна положительность всех ее главных миноров), то все  $d_{ii} = 1$ , и формулы (16) — (19) можно немного упростить.

Расчет по формулам (16) невозможен, если при некотором значении индекса элемент  $s_{kk} = 0$  (в частности,  $s_{11} = 0$  при  $a_{11} = 0$ ). От этого можно избавиться, переставляя на место  $a_{kk}$  другой диагональный элемент  $a_{ii} \neq 0$ , т. е. надо переставить и строки и столбцы, на пересечении которых лежат эти два элемента.

Метод квадратного корня применяют в основном при численном решении интегральных уравнений Фредгольма с симметричным ядром, ибо эта задача сводится к линейной системе с симметричной матрицей, обычно не содержащей нулевых элементов (при регуляризации таких задач симметрия матрицы сохраняется).

**7. Плохо обусловленные системы.** Если система  $Ax = b$  плохо обусловлена, то это значит, что погрешности коэффициентов матрицы и правых частей или погрешности округления при расчетах могут сильно исказить решение. Для уменьшения погрешностей округления можно было бы провести на ЭВМ расчет с двойным или тройным числом знаков. Но при наличии погрешности коэффициентов это бесполезно, и нужно регуляризовать исходную задачу.

Исходную систему (1) можно переписать в эквивалентной форме  $(Ax - b, Ax - b) = 0$ . Если коэффициенты матрицы или правые части известны не точно, то решение также является приближенным. Поэтому на самом деле мы можем требовать только приближенного равенства  $(Ax - b, Ax - b) \approx 0$ . Задача становится неопределенной, и для определенности надо добавить какие-то дополнительные условия.

Таким условием может быть требование, чтобы решение как можно меньше отклонялось от заданного вектора  $x_0$ , т. е. чтобы скалярное произведение  $(x - x_0, x - x_0)$  было минимально. Тогда регуляризованная задача формулируется следующим образом:

$$(Ax - b, Ax - b) + \alpha (x - x_0, x - x_0) = \min, \quad \alpha > 0. \quad (20a)$$

Это можно переписать в эквивалентной форме

$$(x, A^H Ax) - 2(x, A^H b) + (b, b) + \\ + \alpha [(x, x) - 2(x, x_0) + (x_0, x_0)] = \min, \quad (20б)$$

где  $\alpha$  — малый положительный управляющий параметр и  $A^H$  —

эрмитово сопряженная матрица. Варьируя  $\mathbf{x}$  в (20), получим следующее уравнение:

$$(A^H A + \alpha E) \mathbf{x} = A^H \mathbf{b} + \alpha \mathbf{x}_0, \quad (21)$$

где  $E$  — единичная матрица. Решая его (например, методом исключения Гаусса), найдем регуляризованное значение  $\mathbf{x}_\alpha$ , зависящее от параметра  $\alpha$ .

Остановимся на выборе параметра. Если  $\alpha = 0$ , то система (21) переходит в плохо обусловленную систему вида (1). Если же  $\alpha$  велико, то регуляризованная система (21) будет хорошо обусловленной благодаря присутствию в левой части хорошо обусловленной матрицы  $\alpha E$ ; но сама система (21) при большом  $\alpha$  сильно отличается от исходной системы, и регуляризованное решение  $\mathbf{x}_\alpha$  не будет близким к искомому решению. Поэтому слишком малое или слишком большое  $\alpha$  непригодны. Очевидно, оптимальным будет наименьшее значение  $\alpha$ , при котором обусловленность системы (21) еще удовлетворительна.

Для фактического нахождения оптимума вычисляют невязку  $\mathbf{r}_\alpha = A\mathbf{x}_\alpha - \mathbf{b}$  и сравнивают ее по норме с известной погрешностью правых частей  $\delta \mathbf{b}$  и с влиянием погрешности коэффициентов матрицы  $\delta A \cdot \mathbf{x}$ . Если  $\alpha$  слишком велико, то невязка заметно больше этих погрешностей, если слишком мало — то заметно меньше. Проводят серию расчетов с различными  $\alpha$ ; оптимальным считают тот, в котором  $\|\mathbf{r}_\alpha\| \approx \|\delta \mathbf{b}\| + \|\delta A \cdot \mathbf{x}\|$ .

Для выбора  $\mathbf{x}_0$  нужны дополнительные соображения; если их нет, то полагают  $\mathbf{x}_0 = 0$ .

Обоснование изложенного метода дано в главе XIV. Заметим, что матрица системы (21) эрмитова, так что для ее решения можно применять метод квадратного корня.