

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.

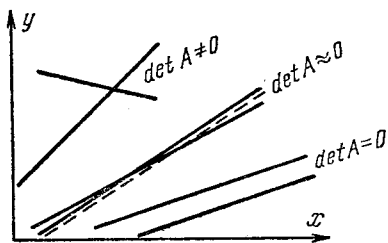


Рис. 24.

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. Дальше 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.

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 $x = A^{-1}b$. 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

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

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 $\det A \approx \approx 0$; 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. Чаше 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 ε 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 $n \lesssim 200$ применяются практически только прямые методы. Итерационные методы приведены в § 3; они выгодны для систем специального вида, со слабо заполненной матрицей очень большого порядка $n \approx 10^3 \div 10^5$. 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

$$\sum_{k=i}^n a_{ik}^{(i)} x_k = b_i^{(i)}, \quad 1 \leq i \leq n, \quad (6)$$

$a_{11}^{(1)}$	$a_{12}^{(1)}$	$a_{13}^{(1)}$...	$a_{1n}^{(1)}$	$b_1^{(1)}$
c_{21}	$a_{22}^{(2)}$	$a_{23}^{(2)}$...	$a_{2n}^{(2)}$	$b_2^{(2)}$
c_{31}	c_{32}	$a_{33}^{(3)}$...	$a_{3n}^{(3)}$	$b_3^{(3)}$
.....					
c_{n1}	c_{n2}	c_{n3}	...	$a_{nn}^{(n)}$	$b_n^{(n)}$

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 $2/3n^3$ 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 α_{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 $3/2n^2$ 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 (δ), 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

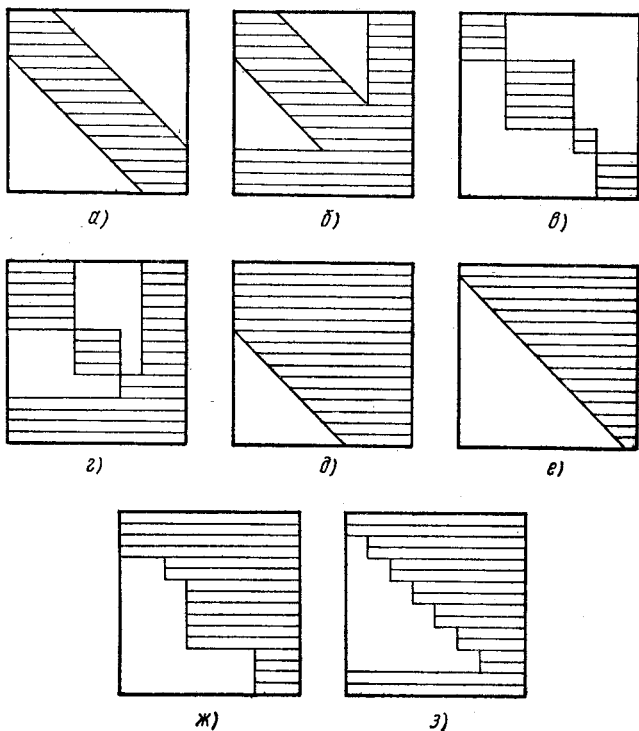


Рис. 25.

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

$$a_i x_{i-1} - b_i x_i + c_i x_{i+1} = d_i, \quad 1 \leq i \leq n, \quad (10)$$

$$a_1 = c_n = 0.$$

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 ξ_{i+1} и η_{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 ξ_1 , η_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 $a_{lk} = a_{ki}^$; эрмитова матрица с вещественными элементами является симметричной.

или окончательно

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

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:

$$S^H z = b, \quad Dy = z, \quad Sx = y, \tag{17}$$

что делается обычным обратным ходом по формулам

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

Определитель матрицы вычисляется по формуле

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

Метод квадратного корня требует примерно $1/3n^3$ 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, $a-g$) 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 $a_{ll} \neq 0$, 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

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

где α — малый положительный управляющий параметр и A^H —

эрмитово сопряженная матрица. Варьируя x в (20), получим следующее уравнение:

$$(A^H A + \alpha E) x = A^H b + \alpha x_0, \quad (21)$$

где E — единичная матрица. Решая его (например, методом исключения Гаусса), найдем регуляризованное значение x_α , зависящее от параметра α .

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

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

Для выбора x_0 нужны дополнительные соображения; если их нет, то полагают $x_0 = 0$.

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

§ 2. Уравнение с одним неизвестным

1. Исследование уравнения. Пусть задана непрерывная функция $f(x)$ и требуется найти все или некоторые корни уравнения

$$f(x) = 0. \quad (22)$$

Эта задача распадается на несколько задач. Во-первых, надо исследовать количество, характер и расположение корней. Во-вторых, найти приближенные значения корней. В-третьих, выбрать из них интересующие нас корни и вычислить их с требуемой точностью.

Первая и вторая задачи решаются аналитическими и графическими методами. Например, многочлен

$$P_n(x) = \sum_{k=0}^n a_k x^k$$

имеет n комплексных корней, не обязательно различных, и все корни лежат внутри круга

$$|x_p| \leq 1 + \frac{1}{|a_n|} \max(|a_0|, |a_1|, \dots, |a_{n-1}|).$$

Правда, эта оценка не оптимальная; модули всех корней могут быть много меньше правой части неравенства, в чем легко

убедиться на примере многочлена $P(x) = \sum_{k=1}^n (x-k)$.

Когда ищутся только действительные корни уравнения, то полезно составить таблицу значений $f(x)$. Если в двух соседних узлах таблицы функция имеет разные знаки, то между этими узлами лежит нечетное число корней уравнения (по меньшей мере один). Если эти узлы близки, то, скорее всего, корень между ними только один. Но выявить по таблице корни четной кратности сложно.

По таблице можно построить график функции $y=f(x)$ и графически найти точки его пересечения с осью абсцисс. Этот способ более нагляден и дает неплохие приближенные значения корней. Во многих задачах техники такая точность уже достаточна. В технике еще популярны графические методы решения уравнений (номография). Построение графика зачастую позволяет выявить даже корни четной кратности.

Иногда удается заменить уравнение (22) эквивалентным ему уравнением $\varphi(x) = \psi(x)$, в котором функции $y_1 = \varphi(x)$ и $y_2 = \psi(x)$ имеют несложные графики. Например, уравнение $x \sin x - 1 = 0$ удобно преобразовать к виду $\sin x = 1/x$. Абсциссы точек пересечения этих графиков будут корнями исходного уравнения.

Приближенные значения корней уточняют различными итерационными методами. Рассмотрим наиболее эффективные из них.

2. Дихотомия (деление пополам). Пусть мы нашли такие точки x_0, x_1 , что $f(x_0)f(x_1) \leq 0$, т. е. на отрезке $[x_0, x_1]$ лежит не менее одного корня уравнения. Найдем середину отрезка $x_2 = (x_0 + x_1)/2$ и вычислим $f(x_2)$. Из двух половин отрезка выберем ту, для которой $f(x_2)f(x_{\text{гран}}) \leq 0$, ибо один из корней лежит на этой половине. Затем новый отрезок опять делим пополам и выберем ту половину, на концах которой функция имеет разные знаки, и т. д. (рис. 26).

Если требуется найти корень с точностью ε , то продолжаем деление пополам до тех пор, пока длина отрезка не станет меньше 2ε . Тогда середина последнего отрезка даст значение корня с требуемой точностью.

Дихотомия проста и очень надежна: к простому корню она сходится для любых непрерывных функций $f(x)$, в том числе недифференцируемых; при этом она устойчива к ошибкам округления. Скорость сходимости невелика: за одну итерацию точность

увеличивается примерно вдвое, т. е. уточнение трех цифр требует 10 итераций. Зато точность ответа гарантируется.

Перечислим недостатки метода. Для начала расчета надо найти отрезок, на котором функция меняет знак. Если в этом отрезке несколько корней, то заранее неизвестно, к какому из них сойдется процесс (хотя к одному из них сойдется). Метод неприменим к корням четной кратности. Для корней нечетной высокой кратности он сходится, но менее точен и хуже устойчив к ошибкам округления, возникающим при вычислении $f(x)$. Наконец, на системы уравнений дихотомия не обобщается.

Дихотомия применяется тогда, когда требуется высокая надежность счета, а скорость сходимости малосущественна.

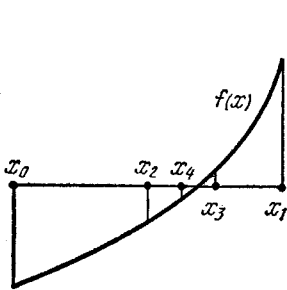


Рис. 26.

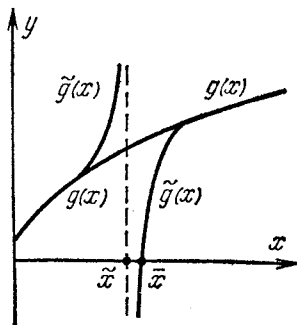


Рис. 27.

3. Удаление корней. Один из недостатков дихотомии — сходимость неизвестно к какому корню — имеется почти у всех итерационных методов. Его можно устранить удалением уже найденного корня.

Если \bar{x}_1 есть простой корень уравнения (22) и $f(x)$ липшиц-непрерывна, то вспомогательная функция $g(x) = f(x)/(x - \bar{x}_1)$ непрерывна, причем все нули функций $f(x)$ и $g(x)$ совпадают, за исключением \bar{x}_1 , ибо $g(\bar{x}_1) \neq 0$. Если \bar{x}_1 — кратный корень уравнения (22), то он будет нулем $g(x)$ кратности на единицу меньше; остальные нули обеих функций по-прежнему будут одинаковы.

Поэтому найденный корень можно удалить, т. е. перейти к функции $g(x)$. Тогда нахождение остальных нулей $f(x)$ сведется к нахождению нулей $g(x)$. Когда мы найдем какой-нибудь корень \bar{x}_2 функции $g(x)$, то этот корень тоже можно удалить, вводя новую вспомогательную функцию $\varphi(x) = g(x)/(x - \bar{x}_2) = f(x)/(x - \bar{x}_1) \times (x - \bar{x}_2)$. Так можно последовательно найти все корни $f(x)$.

Строго говоря, мы находим лишь приближенное значение корня $\tilde{x} \approx \bar{x}$. А функция $\tilde{g}(x) = f(x)/(x - \tilde{x}_1)$ имеет нуль в точке \bar{x}_1 и полюс в близкой к ней точке \tilde{x}_1 (рис. 27); только на некото-

ром расстоянии от этого корня она близка к $g(x)$. Чтобы это не сказывалось при нахождении следующих корней, надо вычислять каждый корень с высокой точностью, особенно если он кратный или вблизи него расположен другой корень уравнения.

Кроме того, в любом методе окончательные итерации вблизи определяемого корня рекомендуется делать не по функциям типа $g(x)$, а по исходной функции $f(x)$. Последние итерации, вычисленные по функции $g(x)$, используются при этом в качестве нулевого приближения. Особенно важно это при нахождении многих корней, ибо чем больше корней удалено, тем меньше нули вспомогательной функции $G(x) = f(x) / \prod_i (x - \tilde{x}_i)$ соответствуют остальным нулям функции $f(x)$.

Учитывая эти предосторожности и вычисляя корни с 8—10 верными десятичными цифрами, зачастую можно определить десятка два корней, о расположении которых заранее ничего не известно (в том числе корней высокой кратности $p \sim 5$).

4. Метод простых итераций. Заменяем уравнение (22) эквивалентным ему уравнением $x = \varphi(x)$. Это можно сделать многими способами, например, положив $\varphi(x) \equiv x + \psi(x)f(x)$, где $\psi(x)$ — произвольная непрерывная знакпостоянная функция. Выберем некоторое нулевое приближение x_0 и вычислим дальнейшие приближения по формулам

$$x_{n+1} = \varphi(x_n), \quad n = 0, 1, 2, \dots \quad (23)$$

Очевидно, если x_n стремится к некоторому пределу \bar{x} , то этот предел есть корень исходного уравнения.

Исследуем условия сходимости. Если $\varphi(x)$ имеет непрерывную производную, тогда

$$x_{n+1} - \bar{x} = \varphi(x_n) - \varphi(\bar{x}) = (x_n - \bar{x}) \varphi'(\xi), \quad (24)$$

где точка ξ лежит между точками x_n и \bar{x} . Поэтому если всюду $|\varphi'(x)| \leq q < 1$, то отрезки $|x_n - \bar{x}|$ убывают не медленней членов геометрической прогрессии со знаменателем $q < 1$ и последовательность x_n сходится при любом нулевом приближении. Если $|\varphi'(\bar{x})| > 1$, то в силу непрерывности $|\varphi'(x)|$ больше единицы и в некоторой окрестности корня; в этом случае итерации не могут сходиться. Если $|\varphi'(\bar{x})| < 1$, но вдали от корня $|\varphi'(x)| > 1$, то итерации сходятся, если нулевое приближение выбрано достаточно близко к корню; при произвольном нулевом приближении сходимости может не быть.

Эти рассуждения переносятся на липшиц-непрерывные функции практически без изменений.

Очевидно, что чем меньше q , тем быстрее сходимость. Вблизи корня асимптотическая сходимость определяется величиной $|\varphi'(\bar{x})|$ и будет особенно быстрой при $\varphi'(\bar{x}) = 0$.

Значит, успех метода зависит от того, насколько удачно выбрано $\varphi(x)$. Например, для извлечения квадратного корня, т. е. решения уравнения $x^2 = a$, можно положить $\varphi(x) = a/x$ или $\varphi(x) = \frac{1}{2} [x + (a/x)]$ и соответственно написать такие итерационные процессы:

$$x_{n+1} = \frac{a}{x_n} \quad \text{или} \quad x_{n+1} = \frac{1}{2} \left(x_n + \frac{a}{x_n} \right). \quad (25)$$

Первый процесс вообще не сходится, а второй сходится при любом $x_0 > 0$; сходится он очень быстро, ибо $\varphi'(x) = 0$. Второй процесс используют при извлечении корня на клавишных машинах.

Каков практический критерий сходимости, т. е. когда надо прекращать итерации (23)? Из (24) видно, что если $\varphi'(x) < 0$, то итерации попеременно оказываются то с одной, то с другой стороны корня, так что корень заключен в интервале (x_n, x_{n-1}) . Это надежная, хотя несколько грубая оценка. Но она неприменима при $\varphi'(x) > 0$, когда итерации сходятся к корню монотонно, т. е. с одной стороны.

Вблизи корня итерации сходятся примерно как геометрическая прогрессия со знаменателем $q = (x_n - x_{n-1}) / (x_{n-1} - x_{n-2})$. Чтобы сумма дальнейших ее членов не превосходила ε , должен выполняться критерий сходимости

$$\left| q \frac{x_n - x_{n-1}}{1 - q} \right| = \frac{(x_n - x_{n-1})^2}{|2x_{n-1} - x_n - x_{n-2}|} < \varepsilon. \quad (26)$$

При выполнении этого условия итерации можно прекращать.

Легко заметить, что выражение в левой части есть поправка Эйткена (4.24). Если последние три простые итерации уточнить процессом Эйткена, то это обычно заметно повышает точность расчета и позволяет ограничиться меньшим числом итераций.

Метод простых итераций и почти все другие итерационные методы имеют важное достоинство: в них не накапливаются ошибки вычислений. Ошибка вычислений эквивалентна некоторому ухудшению очередного приближения. Но это отразится только на числе итераций, а не на точности окончательного результата. Подобные методы устойчивы даже по отношению к грубым ошибкам (сбоям ЭВМ), если только ошибка не выбрасывает очередное приближение за пределы области сходимости.

При обработке эксперимента возникают *стохастические задачи*, где ошибки определения функции велики и носят случайный характер. Погрешность функции δf приводит к погрешности корня $\delta \bar{x} = \delta f(\bar{x}) / f'(\bar{x})$. Однако поскольку ошибки носят случайный характер, то методами статистики можно определить корень гораздо более точно, чем по указанной оценке. Рассмотрим простые итерации

$$x_{n+1} = x_n - a_n f(x_n) \operatorname{sign} f'(x) \quad (27a)$$

при дополнительных условиях

$$a_n > 0, \quad \sum_{n=1}^{\infty} a_n^2 < \infty, \quad \sum_{n=1}^{\infty} a_n = \infty, \quad (27б)$$

которым удовлетворяет, например, последовательность $a_n = (1/n)$. Доказано [47], что $x_n \rightarrow \bar{x}$ при $n \rightarrow \infty$ с вероятностью единица. Использование в формуле (27а) знака производной не означает, что надо вычислять эту производную: достаточно лишь определить ее знак по разности двух значений функции.

Напомним, что стремлением к пределу с вероятностью единица называется сходимость к пределу \bar{x} в подавляющем большинстве случаев (т. е. при разных нулевых приближениях и разных выборах последовательностей a_n), хотя в отдельных случаях процесс может не сходиться или сходиться к другому пределу. Стохастические процессы сходятся медленно, поэтому к детерминированным задачам их целесообразно применять.

5. Метод Ньютона. Он называется также методом *касательных* или методом *линеаризации*. Если x_n есть некоторое приближение к корню \bar{x} , а $f(x)$ имеет непрерывную производную, то уравнение (22) можно преобразовать следующим образом:

$$0 = f(\bar{x}) = f(x_n + (\bar{x} - x_n)) = f(x_n) + (\bar{x} - x_n) f'(\xi).$$

Приблизительно заменяя $f'(\xi)$ на значение в известной точке x_n , получим такой итерационный процесс:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \quad (28)$$

Геометрически этот процесс означает замену на каждой итерации графика $y = f(x)$ касательной к нему (рис. 28).

Метод Ньютона можно рассматривать как частный случай метода простых итераций, если положить $\varphi(x) = x - [f(x)/f'(x)]$. Тогда $\varphi'(x) = (ff''/f'^2)$. Если \bar{x} есть p -кратный корень уравнения (22), то вблизи него $f(x) \approx a(x - \bar{x})^p$; отсюда нетрудно получить $\varphi'(\bar{x}) = (p-1)/p$, т. е. $0 \leq \varphi'(\bar{x}) < 1$. Для простого корня $p=1$ и $\varphi'(\bar{x})=0$. Используя результаты п. 4, можно сформулировать следующие условия сходимости итераций (28). Если нулевое приближение выбрано достаточно близко к корню, ньютоновские итерации сходятся; скорость сходимости велика для простого корня и соответствует скорости геометрической прогрессии для кратного корня. При произвольном нулевом приближении итерации сходятся, если всюду $|ff''| < (f')^2$; в противном случае сходимость будет не при любом нулевом приближении, а только в некоторой окрестности корня.

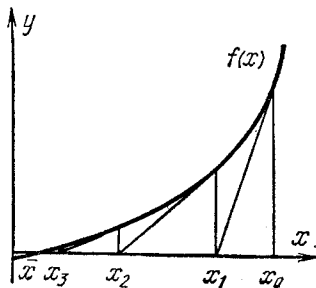


Рис. 28.

Сходимость вблизи любого корня монотонна, что легко видеть из рис. 28; но вдали от корня возможна немонотонность итераций. Отметим, что рис. 28 указывает еще на одно достаточное

условие сходимости итераций. Пусть $f''(x) \geq 0$ справа от корня на отрезке $[\bar{x}, a]$; если x_0 выбрано также справа от корня на этом отрезке, то итерации (28) сходятся, причем монотонно. То же будет, если $f''(x) \leq 0$ слева от корня на отрезке $[b, \bar{x}]$, и на этом же отрезке выбрано нулевое приближение. Таким образом, итерации сходятся к корню с той стороны, с которой $f(x) f''(x) \geq 0$.

Оценим скорость сходимости вблизи простого корня. По определению простых итераций, $\bar{x} - x_n = \varphi(\bar{x}) - \varphi(x_{n-1})$. Разлагая правую часть по формуле Тейлора и учитывая равенство $\varphi'(\bar{x}) = 0$, получим

$$x_n - \bar{x} = \frac{1}{2} (x_{n-1} - \bar{x})^2 \varphi''(\xi), \quad \xi \in (x_{n-1}, \bar{x}), \quad (29)$$

т. е. погрешность очередного приближения примерно равна квадрату погрешности предыдущего приближения. Например, если $(n-1)$ -я итерация давала 3 верных знака, то n -я даст примерно 6 верных знаков, а $(n+1)$ -я — примерно 12 знаков. Это иллюстрирует быструю сходимость вблизи корня. Разумеется, вдали от корня подобные соображения неприменимы.

Таблица 16

$$f(x) \equiv x^2 - 4 = 0$$

n	x_n , метод Ньютона	x_n , метод секущих
0	1,0000	1,0000
1	2,5000	2,5000
3	2,0500	1,8571
4	2,0001	1,9836

Если вычисляется корень высокой кратности, то $f'(x)$ в знаменателе формулы (28) становится малой вблизи корня. Чтобы не было потери точности, отношение $f(x)/f'(x)$ надо вычислять достаточно аккуратно. К остальным погрешностям расчета метод Ньютона хорошо устойчив.

Для нахождения корней произвольной дифференцируемой функции чаще всего применяют метод Ньютона, особенно если известны разумные начальные приближения для корней.

Пример. Рассмотрим решение уравнения $f(x) \equiv x^2 - a = 0$. Тогда общая формула метода Ньютона (28) принимает вид

$$x_{n+1} = x_n - \frac{x_n^2 - a}{2x_n} = \frac{1}{2} \left(x_n + \frac{a}{x_n} \right).$$

Мы получили вторую формулу (25), которая, как отмечалось раньше, позволяет очень быстро находить квадратный корень с помощью только сложения и деления. Для иллюстрации в таблице 16 приведен ход итераций при извлечении квадратного корня из $a=4$. Видно, что сходимость очень быстрая; несмотря на неважное нулевое приближение, уже третья итерация дает точность 0,005%. Попутно можно заметить, что вблизи корня итерации сходятся с одной стороны, т. е. монотонно, хотя первая итерация дает переброс на другую сторону корня.

6. Процессы высоких порядков. В методе простых итераций выберем функцию $\varphi(x)$ так, чтобы выполнялось

$$\varphi'(\bar{x}) = \varphi''(\bar{x}) = \dots = \varphi^{(p-1)}(\bar{x}) = 0, \quad \varphi^{(p)}(\bar{x}) \neq 0. \quad (30)$$

Итерационный процесс (23) с такой функцией называют *стационарным процессом p -го порядка*. Скорость сходимости этого процесса вблизи корня можно получить из следующих равенств:

$$x_{n+1} - \bar{x} = \varphi(x_n) - \varphi(\bar{x}) = \frac{1}{p!} (x_n - \bar{x})^p \varphi^{(p)}(\xi), \quad \xi \in (x_n, \bar{x}). \quad (31a)$$

Если $|\varphi^{(p)}(x)| \leq M_p$, то отсюда следует

$$|x_n - \bar{x}| \leq (M_p/p!)^{(p^n - 1)/(p - 1)} |x_0 - \bar{x}|^{p^n}. \quad (31b)$$

Сходимость при $p=1$ называют линейной (это собственно метод простых итераций), при $p=2$ — квадратичной (например, метод Ньютона), а при $p=3$ — кубической. Очевидно, чем больше p , тем быстрее сходятся итерации вблизи корня; к сожалению, тем меньше область гарантированной сходимости этих приближений.

Фактически у процессов высокого порядка выход на их асимптотическую скорость сходимости (31) обычно наступает только тогда, когда итерации уже почти сошлись, т. е. для получения всех верных разрядов числа осталось сделать одну — три итерации. Поэтому такие процессы (за исключением метода парабол) редко употребляются.

7. Метод секущих *) [48]. В методе Ньютона требуется вычислять производную функции, что не всегда удобно. Можно заменить производную первой разделенной разностью, найденной по двум последним итерациям, т. е. заменить касательную секущей. Тогда вместо процесса (28) получим

$$x_{n+1} = x_n - \frac{(x_n - x_{n-1})f(x_n)}{f(x_n) - f(x_{n-1})}. \quad (32)$$

Для начала процесса надо задать x_0 и x_1 (рис. 29). Такие процессы, где для вычисления очередного приближения надо знать два предыдущих, называют *двухшаговыми*.

Эти, казалось бы, небольшие изменения сильно влияют на характер итераций. Например, сходимость итераций может быть немонотонной не только вдали от корня, но и в малой окрестности корня. Скорость сходимости также изменяется. Иллюстрацией служит приведенный в таблице 16 расчет по методу секущих; для удобства сравнения с методом Ньютона первые два

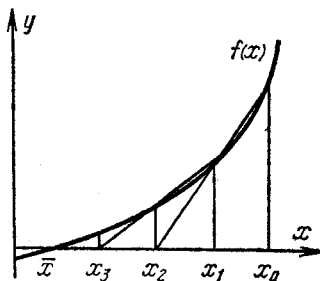


Рис. 29.

*) В математической литературе это название нередко употребляют для другого метода, который по его геометрической интерпретации следовало бы называть методом хорд.

приближения взяты одинаковыми. Видно, что метод секущих сходится медленнее.

Скорость сходимости можно оценить, разлагая все функции в (32) по формуле Тейлора с центром \bar{x} . Получим с точностью до бесконечно малых более высокого порядка

$$x_{n+1} - \bar{x} = a(x_n - \bar{x})(x_{n-1} - \bar{x}), \quad a = \frac{f''(\bar{x})}{2f'(\bar{x})}. \quad (33)$$

Решение этого рекуррентного соотношения естественно искать в виде $x_{n+1} - \bar{x} = a^\alpha (x_n - \bar{x})^\beta$, аналогичном (29) или (31a). Подставляя эту форму в соотношение (33), получим

$$\alpha\beta = 1, \quad \beta^2 - \beta - 1 = 0. \quad (34)$$

Только положительный корень β квадратного уравнения (34) соответствует убыванию ошибки, т. е. сходящемуся процессу. Следовательно, в методе секущих

$$x_{n+1} - \bar{x} = a^{1/\beta} (x_n - \bar{x})^\beta, \quad \beta = \frac{1}{2}(\sqrt{5} + 1) \approx 1,62, \quad (1/\beta) \approx 0,62, \quad (35)$$

в то время как в методе Ньютона ошибка убывает быстрее (соответствуя $\beta = 2$). Но в методе Ньютона на каждой итерации надо вычислять и функцию, и производную, а в методе секущих — только функцию. Поэтому при одинаковом объеме вычислений в методе секущих можно сделать вдвое больше итераций и получить более высокую точность.

В знаменателе формулы (32) стоит разность значений функции. Вдали от корня это несущественно; но вблизи корня, особенно корня высокой кратности, значения функции малы и очень близки. Возникает потеря значащих цифр, приводящая к «разболтке» счета. Это ограничивает точность, с которой можно найти корень; для простых корней это ограничение невелико, а для кратных может быть существенным.

Заметим, что приводить формулу (32) к общему знаменателю не следует: увеличится потеря точности в расчетах.

От «разболтки» боятся так называемым приемом Гарвика. Выбирают не очень малое ε , ведут итерации до выполнения условия $|x_{n+1} - x_n| < \varepsilon$ и затем продолжают расчет до тех пор, пока $|x_{n+1} - x_n|$ убывают. Первое же возрастание обычно означает начало «разболтки»; тогда расчет прекращают и последнюю итерацию не используют.

8. Метод парабол. Метод секущих можно рассматривать как замену функции $f(x)$ интерполяционным многочленом первой степени, проведенным по узлам x_n, x_{n-1} . По трем последним

итерациям можно построить интерполяционный многочлен второй степени, т. е. заменить график функции параболой. Запишем интерполяционный многочлен в форме Ньютона

$$\mathcal{P}_2(x) = f(x_n) + (x - x_n)f(x_n, x_{n-1}) + (x - x_n)(x - x_{n-1})f(x_n, x_{n-1}, x_{n-2}).$$

Приравнявая его нулю, получим квадратное уравнение

$$az^2 + bz + c = 0, \quad (36a)$$

где

$$\begin{aligned} z &= x - x_n, & a &= f(x_n, x_{n-1}, x_{n-2}), \\ b &= a(x_n - x_{n-1}) + f(x_n, x_{n-1}), & c &= f(x_n). \end{aligned} \quad (36b)$$

Тот из двух корней квадратного уравнения (36), который меньше по модулю, определяет новое приближение $x_{n+1} = x_n + z$.

Очевидно, для начала расчета надо задать три первых приближения x_0, x_1, x_2 (обычно наугад выбирают три числа), т. е. процесс является *трехшаговым*.

Метод парабол построен по образцу методов третьего порядка. Однако замена производных разделенными разностями приводит к существенному уменьшению скорости сходимости. Рассуждениями, аналогичными рассуждениям в п. 7, можно показать, что вблизи простого корня выполняется соотношение

$$|x_{n+1} - \bar{x}| \approx \left| \frac{f'''(\bar{x})}{6f'(\bar{x})} \right|^{0,42} |x_n - \bar{x}|^{1,84}, \quad (37)$$

т. е. сходимость даже медленнее квадратичной. Вблизи кратного корня сходимость еще медленнее (хотя и более быстрая, чем линейная). Заметим, что строить аналогичные методы с использованием интерполяционного многочлена еще более высокой степени невыгодно: сходимость все равно будет медленней квадратичной, а расчет сильно усложняется.

В методе парабол «разболтка» счета вблизи корня сказывается еще сильнее, чем в методе секущих, ибо в расчете участвуют вторые разности. Тем не менее корни можно найти с хорошей точностью; для определения оптимального числа итераций удобно пользоваться приемом Гарвика, описанном в п. 7.

Метод парабол имеет важное достоинство. Даже если все предыдущие приближения действительны, уравнение (36) может привести к комплексным числам. Поэтому процесс может естественно сойтись к комплексному корню исходного уравнения. В методах простых итераций, касательных или секущих для сходимости к комплексному корню может потребоваться задание комплексного начального приближения (если $f(x)$ вещественна при вещественном аргументе).

Корни многочлена. Метод парабол оказался исключительно эффективным для нахождения всех корней многочлена высокой степени. Если $f(x)$ — алгебраический многочлен, то, хотя

сходимость метода при произвольном начальном приближении и не доказана, на практике итерации всегда сходятся к какому-нибудь корню, причем быстро. Для многочлена частное $f(x)/(x - \bar{x})$ есть тоже многочлен; поэтому последовательно удаляя найденные корни, можно найти все корни исходного многочлена.

Замечание 1. Если $f(x)$ — многочлен высокой степени, то возникают дополнительные трудности. Многочлен быстро возрастает при увеличении аргумента, поэтому в программе для ЭВМ должна быть страховка от переполнения. Обычно вводят масштабные множители, величина которых связана с диапазоном изменения аргумента.

Замечание 2. Наибольшие по модулю корни многочлена высокой степени могут быть очень чувствительны к погрешности коэффициентов при старших степенях. Например, корнями многочлена

$$P_{20}(x) = \prod_{k=1}^{20} (x - k)$$

являются последовательные целые числа $\bar{x}_k = 1, 2, \dots, 20$. А слегка измененный многочлен $\tilde{P}_{20}(x) = P_{20}(x) - 2^{-23}x^{19}$ имеет такие корни:

$$1,0; 2,0; \dots; 8,0; 8,9; 10,1 \pm 0,6i; \dots; 19,5 \pm 1,9i; 20,8$$

(здесь приведен только один знак после запятой). Кратные или близкие корни могут быть слабо устойчивыми даже при меньших степенях многочлена.

Замечание 3. Для удаления вычисленных корней надо делить многочлен. Это вносит погрешность округления в коэффициенты и влияет на точность нахождения следующих корней. На практике отмечено, что если сначала удалять меньшие по модулю корни, точность падает мало, но если начать удаление с больших корней, точность может упасть катастрофически. Поэтому за начальное приближение берут $x_0 = -1$, $x_1 = +1$. $x_2 = 0$; тогда итерации обычно сходятся к наименьшему по модулю корню. Его удаляют и по такому же начальному приближению ищут следующий корень и т. д. При такой организации вычислений потеря точности будет небольшой.

9. Метод квадрирования. Этот метод позволяет найти все корни многочлена. Запишем многочлен n -й степени двумя способами:

$$P_n(x) = \sum_{k=0}^n a_k x^k = a_n \prod_{l=1}^n (x - \bar{x}_l), \quad (38)$$

где \bar{x}_l — корни многочлена. Сравнивая обе формы записи, получим равенства

Виета

$$\sum_{k=1}^n \bar{x}_k = -\frac{a_{n-1}}{a_n}, \quad \sum_{k>l} \bar{x}_k \bar{x}_l = +\frac{a_{n-2}}{a_n}, \quad \sum_{k>l>m} \bar{x}_k \bar{x}_l \bar{x}_m = -\frac{a_{n-3}}{a_n}, \dots \quad (39)$$

Предположим, что корни многочлена сильно различаются по абсолютной величине: $|\bar{x}_1| \gg |\bar{x}_2| \gg \dots \gg |\bar{x}_n|$. Тогда из равенств Виета получаются приближенные значения корней

$$\bar{x}_1 \approx -\frac{a_{n-1}}{a_n}, \quad \bar{x}_2 \approx -\frac{a_{n-2}}{a_{n-1}}, \quad \dots, \quad \bar{x}_n \approx -\frac{a_0}{a_1}. \quad (40)$$

Если модули корней мало различаются, то эти формулы слишком неточны. Но квадраты модулей будут различаться сильнее, чем сами модули. Поменяем в многочлене (38) знаки всех корней, что эквивалентно смене знака у всех нечетных коэффициентов. Умножая измененный многочлен на исходный, получим

$$\left[\sum_{k=1}^n a_k x^k \right] \cdot \left[\sum_{l=1}^n (-1)^{n-l} a_l x^l \right] = a_n^2 \prod_{m=1}^n (x^2 - \bar{x}_m^2) \equiv Q_n(z), \quad z = x^2. \quad (41)$$

Многочлен $Q_n(z)$ имеет n -ю степень и называется *квадрированным многочленом*. Его корни равны квадратам корней исходного многочлена. Нахождение квадрированного многочлена сравнительно трудоемко; его коэффициенты можно вычислять по формулам

$$b_l = \sum_{m=-\alpha}^{\alpha} (-1)^{l-m} a_{l+m} a_{l-m}, \quad \alpha = \min(l, n-l). \quad (42)$$

Для фактического выполнения этих вычислений удобно записать произведения коэффициентов с нужными знаками в форме таблицы 17; знаки ставятся в шахматном порядке. Произведения суммируются по косым строкам, как указано стрелками.

Таблица 17

	a_0	a_1	a_2	...	a_n
a_0	$a_0 a_0$	$-a_0 a_1$	$a_0 a_2$...	$\pm a_0 a_n$
a_1	$-a_1 a_0$	$a_1 a_1$	$-a_1 a_2$...	$\mp a_1 a_n$
a_2	$a_2 a_0$	$-a_2 a_1$	$a_2 a_2$...	$\pm a_2 a_n$
...
a_n	$\pm a_n a_0$	$\mp a_n a_1$	$\pm a_n a_2$...	$a_n a_n$

Для квадрированного многочлена корни по формулам (40), где вместо a_k надо вставить b_k , определяются точнее. Если результат будет мало отличаться от предыдущего, то на нем можно остановиться. Если отличие заметное, то квадрирование надо повторить. Повторяя квадрирование много раз, получим

быстро сходящийся итерационный процесс (можно показать, что его сходимость будет квадратичной).

При различающихся по модулю корнях после многократного квадрирования выполняются соотношения

$$\left| \frac{b_{n-1}}{b_n} \right| \gg \left| \frac{b_{n-2}}{b_{n-1}} \right| \gg \dots \gg \left| \frac{b_0}{b_1} \right|.$$

Если среди корней есть равные по модулю (в частности, кратные), то это соотношение нарушается. Например, если второй корень будет двукратным, то получим

$$\left| \frac{b_{n-1}}{b_n} \right| \gg \left| \frac{b_{n-2}}{b_{n-1}} \right| \sim \left| \frac{b_{n-3}}{b_{n-2}} \right| \gg \left| \frac{b_{n-4}}{b_{n-3}} \right| \gg \dots$$

Если корни равны только по модулю, но $\bar{x}_k \neq \bar{x}_l$ (например, комплексно-сопряженные корни), то это случайное совпадение. Такие корни удобнее всего разделять сдвигом. Рассмотрим многочлен $R(x) = P_n(x - \xi)$, где ξ — случайно выбранное комплексное число. Корни многочлена $R(x)$ будут уже не равны между собой по модулю, ибо они отличаются от корней исходного многочлена на комплексную величину ξ .

Кратные корни разделить сдвигом нельзя. Для них надо составлять специальные формулы, явно учитывающие, какой корень какую кратность имеет. Определить кратность корней можно по поведению отношений соседних коэффициентов.

Обратный переход от корней квадрированных уравнений к корням исходного уравнения осуществляется последовательным извлечением квадратного корня. При этом появляются ложные корни; их надо обнаружить подстановкой в исходное уравнение и отбросить.

Метод квадрирования позволяет легко выполнить все расчеты на клавишных машинах. Он не требует задания какого-либо нулевого приближения. Но для программирования на ЭВМ этот метод не особенно удобен. Во-первых, после нескольких квадрирований в расчете возникают обычно большие числа, и возможно переполнение, от которого приходится страховать введением масштабных множителей. Во-вторых, при наличии кратных корней требуется произвести довольно громоздкий логический анализ и применить нестандартные формулы вычисления.

§ 3. Системы нелинейных уравнений

1. Метод простых итераций. Систему нелинейных уравнений можно кратко записать в векторном виде

$$f(x) = 0 \quad (43)$$

или более подробно в координатном виде

$$f_k(x_1, x_2, \dots, x_n) = 0, \quad 1 \leq k \leq n.$$

Такие системы решают практически только итерационными методами. Нулевое приближение в случае двух переменных можно найти графически: построить на плоскости (x_1, x_2) кривые $f_1(x_1, x_2) = 0$ и $f_2(x_1, x_2) = 0$ и найти точки их пересечения. Для трех и более переменных (а также для комплексных корней) удовлетворительных способов подбора нулевых приближений нет.

Рассмотрим метод *простых итераций*, называемый также методом *последовательных приближений*. Аналогично одномерному

случаю, заменим нелинейную систему (43) эквивалентной системой специального вида $\mathbf{x} = \Phi(\mathbf{x})$. Выберем некоторое нулевое приближение и дальнейшие приближения найдем по формулам

$$\begin{aligned} & \mathbf{x}^{(s+1)} = \Phi(\mathbf{x}^{(s)}) \\ \text{или} & \quad x_k^{(s+1)} = \varphi_k(x_1^{(s)}, x_2^{(s)}, \dots, x_n^{(s)}), \quad 1 \leq k \leq n. \end{aligned} \quad (44)$$

Если итерации сходятся, то они сходятся к решению уравнения (предполагается, что решение существует).

Сходимость итераций исследуем так же, как для одной переменной. Обозначим компоненты решения через \bar{x}_k и преобразуем погрешность очередной итерации

$$\begin{aligned} x_k^{(s+1)} - \bar{x}_k &= \varphi_k(x_1^{(s)}, \dots, x_n^{(s)}) - \varphi_k(\bar{x}_1, \dots, \bar{x}_n) = \\ &= \varphi_k(\mathbf{x}^{(s)}) - \varphi_k(\bar{\mathbf{x}}) = [\partial\varphi_k(\xi_k)/\partial l]_{\rho}(\mathbf{x}^{(s)}, \bar{\mathbf{x}}) = \\ &= \sum_{i=1}^n (x_i^{(s)} - \bar{x}_i) [\partial\varphi_k(\xi_k)/\partial x_i], \end{aligned}$$

где l — направление, соединяющее многомерные точки $\mathbf{x}^{(s)}$ и $\bar{\mathbf{x}}$, а ξ_k — некоторая точка, лежащая между ними на этом направлении. Это равенство означает, что вектор погрешности нового приближения равен матрице производных, умноженной на вектор погрешности предыдущего приближения. Если какая-нибудь норма матрицы производных $\{\partial\varphi_k(\xi_k)/\partial x_i\}$, согласованная с некоторой нормой вектора, меньше единицы, то норма погрешности убывает от итерации к итерации по геометрической прогрессии. Это означает линейную сходимость метода.

На практике удобнее рассматривать матрицу с элементами $M_{ki} = \max |\partial\varphi_k/\partial x_i|$. Нормы этой матрицы мажорируют соответствующие нормы матрицы производных, поэтому *достаточным условием сходимости является* $\|M_{ki}\| < 1$. При использовании разных норм матриц это условие принимает такие формы:

$$\sum_{i=1}^n M_{ki} < 1, \quad \text{или} \quad \sum_{k=1}^n M_{ki} < 1, \quad \text{или} \quad \sum_{k,i=1}^n M_{ki}^2 < 1. \quad (45)$$

Каждая норма матрицы согласована с определенной нормой вектора, но в конечномерном пространстве все нормы эквивалентны. Значит, если итерации сходятся в одной норме, то они сходятся и во всех остальных нормах.

Поскольку сходимость линейная, то оканчивать итерации можно по критерию сходимости (26), выполнение которого надо проверять для каждой компоненты. Линейная сходимость довольно медленна; поэтому полезно уточнять результат процессом Эйткена по трем последним итерациям.

Сами вычисления в методе последовательных приближений просты. Но зато сложно найти такую систему $\mathbf{x} = \Phi(\mathbf{x})$, которая была бы эквивалентна исходной системе $\mathbf{f}(\mathbf{x}) = 0$ и одновременно обеспечивала бы сходимость.

Сходимость метода нередко удается улучшить. В методе простых итераций найденное приближение $x_k^{(s+1)}$ используется только для вычисления следующей итерации. Можно использовать его уже на данной итерации для вычисления следующих компонент:

$$x_k^{(s+1)} = \Phi_k(x_1^{(s+1)}, x_2^{(s+1)}, \dots, x_{k-1}^{(s+1)}, x_k^{(s)}, x_{k+1}^{(s)}, \dots, x_n^{(s)}). \quad (46)$$

Сходимость этого варианта метода тоже линейная; детально ее исследовать не будем. При ручных расчетах можно еще ускорить сходимость за счет перестановок отдельных уравнений на основе анализа их невязок $r_k^{(s)} = x_k^{(s)} - \Phi_k(x^{(s)}) = x_k^{(s)} - x_k^{(s+1)}$; но для расчетов на ЭВМ это неудобно, ибо обычно такой анализ полуинтуитивен и плохо алгоритмируется.

2. Метод Ньютона. Пусть известно некоторое приближение $\mathbf{x}^{(s)}$ к корню $\bar{\mathbf{x}}$. Как и для одной переменной, запишем исходную систему (43) в виде $\mathbf{f}(\mathbf{x}^{(s)} + \Delta\mathbf{x}) = 0$, где $\Delta\mathbf{x} = \bar{\mathbf{x}} - \mathbf{x}^{(s)}$. Разлагая эти уравнения в ряды и ограничиваясь первыми дифференциалами, т. е. линеаризуя функцию, получим

$$\sum_{i=1}^n \frac{\partial f_k(x^{(s)})}{\partial x_i} \Delta x_i^{(s)} = -f_k(x^{(s)}), \quad 1 \leq k \leq n. \quad (47)$$

Это система уравнений, линейных относительно приращений $\Delta x_i^{(s)}$; все коэффициенты этой системы выражаются через последнее приближение $\mathbf{x}^{(s)}$. Решив эту систему (например, методом исключения), найдем новое приближение $\mathbf{x}^{(s+1)} = \mathbf{x}^{(s)} + \Delta\mathbf{x}^{(s)}$.

Как и для одной переменной, метод Ньютона можно формально свести к методу последовательных приближений, положив $\Phi(\mathbf{x}) = \mathbf{x} - [\partial\mathbf{f}/\partial\mathbf{x}]^{-1} \mathbf{f}(\mathbf{x})$, где $[\partial\mathbf{f}/\partial\mathbf{x}]^{-1}$ есть матрица, обратная матрице производных. Аналогично проводится теоретический анализ условий сходимости. Однако достаточное условие сходимости, записанное в координатной форме, здесь имеет настолько сложный вид, что проверить его выполнимость почти никогда не удастся. Отметим только очевидный результат: *в достаточно малой окрестности корня итерации сходятся, если $\det[\partial\mathbf{f}/\partial\mathbf{x}] \neq 0$, причем сходимость квадратичная.*

Следовательно, если нулевое приближение выбрано удачно, то метод Ньютона сходится, причем очень быстро (обычно за 3—5 итераций). Поэтому на практике этот метод используют чаще всего.

В отличие от метода простых итераций, для метода Ньютона хорошим критерием окончания итераций является условие

$\|x^{(s)} - x^{(s+1)}\| \leq \varepsilon$. В самом деле, вблизи корня ньютоновские итерации сходятся квадратично, поэтому если этот критерий выполнен, то $\|x^{(s+1)} - x\| \approx \varepsilon^2 \ll \varepsilon$. Выбирая $\varepsilon \approx 10^{-5} - 10^{-6}$, можно получить решение с десятком верных знаков.

Вычисления в методе Ньютона несколько сложнее, чем при простых итерациях, ибо на каждой итерации требуется находить матрицу производных и решать систему линейных уравнений. Поэтому в некоторых учебниках рекомендуют такой прием: вычислить матрицу $[df/dx]^{-1}$ только на начальной итерации и использовать ее на всех остальных итерациях.

Однако сходимость при этом видоизменении становится линейной, причем обычно не с малой константой, ибо матрица производных на начальной итерации может заметно отличаться от окончательной. Поэтому скорость сходимости заметно уменьшается и требуемое число итераций возрастает.

3. Методы спуска. Рассмотрим функцию $\Phi(x) = \sum_{k=1}^n |f_k(x)|^2$.

Она неотрицательна и обращается в нуль в том и только в том случае, если $f(x) = 0$. Таким образом, решение исходной системы уравнений (43) будет одновременно нулевым минимумом скалярной функции многих переменных $\Phi(x)$.

Иногда бывает проще искать такой минимум, чем решать систему уравнений. Методы поиска минимума будут рассмотрены в главе VII. В основном это итерационные методы спуска, т. е. движения в направлении убывания функции. Все методы спуска для гладких функций сходятся, но зачастую — довольно плохо. Поэтому на хорошую точность полученного решения трудно рассчитывать; однако этим способом обычно можно найти разумное приближение, которое потом можно уточнять методом Ньютона.

Надо помнить, что метод спуска в зависимости от выбора нулевого приближения может сойтись к любому минимуму функции. А функция $\Phi(x)$ может иметь ненулевые локальные минимумы, которые не являются решениями исходной системы уравнений.

4. Итерационные методы решения линейных систем иногда дополняют, а иногда заменяют прямые методы.

Решая линейную систему (1) общего вида методом исключения, попутно можно проверить, насколько хорошо она обусловлена. Решение, найденное прямым методом, из-за ошибок округления будет приближенным. Нетрудно проверить, что поправки к нему удовлетворяют уравнениям

$$\sum_{i=1}^n a_{ki} \Delta x_i = r_k, \quad 1 \leq k \leq n, \quad (48)$$

где r_k — невязки (7). Это линейная система с той же матрицей, что исходная система (1). Решим ее, используя ранее найденные коэффициенты c_{mk} (т. е. почти не увеличивая общего объема вычислений). Ранее отмечалось, что в методе Гаусса невязки малы. Если величины Δx_i тоже окажутся малыми, то система хорошо обусловлена; если большими — то плохо. В последнем случае зачастую удастся уточнить решение, рассматривая (48) как итерационный процесс Ньютона и делая 2—3 итерации.

Для уточнения обратной матрицы тоже есть итерационный процесс с квадратичной сходимостью

$$A_s^{-1} + 1 = A_s^{-1} + A_s^{-1}R_s, \quad R_s = E - AA_s^{-1}. \quad (49)$$

Однако каждая итерация этого процесса требует $4n^3$ арифметических действий, т. е. вдвое больше, чем прямое обращение матрицы по методам Гаусса или Жордана. Поэтому в практических вычислениях этот процесс теперь не применяют.

При очень плохой обусловленности матрицы оба описанных метода уточнения могут потребовать вычислений с двойным и более числом знаков, но тогда лучше применять регуляризирующие алгоритмы.

Есть важная группа задач, приводящая к линейным системам с сотнями и тысячами неизвестных. Это решение двумерных и трехмерных уравнений в частных производных эллиптического типа при помощи разностных схем. Матрицы таких систем слабо заполнены, но расположение нулевых элементов таково, что метод исключения не может полностью использовать особенности структуры матрицы и приводит к большому объему вычислений. Кроме того, в методе исключения матрицы таких систем не помещаются в оперативной памяти ЭВМ, а обращение к внешней памяти еще более увеличивает время расчета.

Подобные линейные системы зачастую выгодно решать итерационными методами. Современные итерационные методы мы рассмотрим в главе XII, посвященной эллиптическим уравнениям. Здесь упомянем только два старых метода, уже не используемых в практических вычислениях, но удобных для некоторых теоретических оценок.

Один из них — стационарный метод простых итераций, основанный на приведении системы $Ax = b$ к эквивалентной системе специального вида:

$$x = Cx + d. \quad (50)$$

Запись итерационного процесса очевидна. Для сходимости итераций достаточно, если какая-нибудь норма $\|C\| < 1$. В этой задаче известно даже необходимое и достаточное условие сходимости — модули всех собственных значений матрицы C должны быть меньше единицы; но на практике этим условием трудно воспользоваться, ибо найти собственные значения обычно сложнее, чем решить линейную систему.

К виду (50) систему можно привести, например, выделением диагональных элементов

$$x_k = \frac{1}{a_{kk}} \left(b_k - \sum_{i \neq k} a_{ki} x_i \right), \quad 1 \leq k \leq n. \quad (51)$$

В этой записи легко учесть наличие нулей в матрице A и при умножении матрицы на вектор суммировать только по ненулевым элементам. При использовании различных норм матрицы достаточные условия сходимости итераций принимают вид

$$\sum_{i \neq k} \left| \frac{a_{ki}}{a_{kk}} \right| < 1, \quad \text{или} \quad \sum_{k \neq i} \left| \frac{a_{ki}}{a_{kk}} \right| < 1, \quad \text{или} \quad \sum_k \sum_{i \neq k} \left| \frac{a_{ki}}{a_{kk}} \right|^2 < 1, \quad (52)$$

что означает преобладание диагональных элементов матрицы (сравните условия устойчивости метода прогонки (14)).

Если метод сходится, то корень уравнения существует. Таким образом, преобладание диагональных элементов матрицы в смысле одного из неравенств (52) является признаком того, что система линейных уравнений (1) имеет решение, т. е. $\det A \neq 0$. Этим признаком мы будем часто пользоваться при исследовании разрешимости неявных разностных схем.

Заметим, что чем меньше $\|C\|$, тем быстрее сходятся итерации. Наилучшие современные методы основаны на специальных способах выбора матрицы C , при которых ее норма становится минимальной в некотором смысле.

Второй — метод Зейделя. Для уравнения (51) он имеет такой вид:

$$x_k^{(s+1)} = \frac{1}{a_{kk}} \left(b_k - \sum_{i=1}^{k-1} a_{ki} x_i^{(s+1)} - \sum_{i=k+1}^n a_{ki} x_i^{(s)} \right), \quad 1 \leq k \leq n. \quad (53)$$

Необходимое и достаточное условие его сходимости не совпадает с условием сходимости простых итераций, и в разных условиях он может быть выгоден или невыгоден. Отметим один признак сходимости: если матрица A — эрмитова и положительно определенная, то метод Зейделя в форме (53) совпадает с методом покоординатного спуска для решения задачи на минимум квадратичной формы $(x, Ax) - 2(b, x) = \min$; как будет показано в главе VII, метод покоординатного спуска сходится, что обеспечивает сходимость метода Зейделя в данном случае.

ЗАДАЧИ

1. Записать для почти треугольной матрицы (рис. 25, д) формулы метода исключения Гаусса с обходом нулевых элементов.

2. Показать, что при преобразовании эрмитовых матриц, изображенных на рис. 25, а—г, в произведение (15) структура треугольных матриц подобна структуре исходных матриц.

3. Доказать, что первый итерационный процесс (25) не сходится, а второй сходится при любом (положительном) нулевом приближении.

4. Найти асимптотическую скорость сходимости метода секущих (32) вблизи корня кратности p .

5. Доказать, что скорость сходимости метода парабол вблизи простого корня определяется формулой (37); исследовать сходимость вблизи кратного корня.

6. Доказать, что метод квадрирования сходится квадратично.

7. Вывести формулы метода квадрирования для случая, когда наибольший по модулю корень — двукратный.

8. Доказать, что итерационный процесс (49) для нахождения обратной матрицы сходится квадратично.