ГЛАВА 1

Прямые методы решения систем линейных уравнений

Многие задачи практики приводят к необходимости решать системы линейных алгебраических уравнений (СЛАУ). При конструировании инженерных сооружений, приборов, обработке результатов измерений, решении задач планирования производственного процесса и многих других задач техники, экономики, научного эксперимента приходится решать СЛАУ.

Совокупность линейных соотношений

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

как мы знаем, называется системой линейных алгебраических уравнений. Компактно эти уравнения записываются в виде[1)]

$$\sum_{j=1}^{n} a_{ij}\, x_j = b_i, \quad i = 1 : n,$$

а также в матричном виде

$$Ax = b.$$

Здесь

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

[1)] Запись $i = 1 : n$ означает, что $i = 1, 2, \ldots, n$.

Матрица коэффициентов A и вектор правой части b системы уравнений (столбец свободных членов) являются данными, а вектор неизвестных x требуется определить.

Исследование ряда научно-технических, экономических и прочих проблем приводит к математическим моделям непосредственно в форме систем линейных алгебраических уравнений. Однако гораздо чаще они появляются в процессе математического моделирования как промежуточный этап при решении более сложной задачи, например, после дискретизации (и, если необходимо, линеаризации) интегральных, дифференциальных, интегро-дифференциальных уравнений или систем уравнений такого сорта. В силу этого задачи линейной алгебры являются наиболее часто решаемыми математическими задачами в процессе математического моделирования.

1. Разрешимость СЛАУ. Итак, рассмотрим СЛАУ

$$\sum_{j=1}^{n} a_{ij}\, x_j = b_i, \quad i = 1 : n,$$

которая в матричном виде записывается как

$$Ax = b. \tag{1}$$

Далее мы будем рассматривать только случай, когда данные задачи являются вещественными (т.е. $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$), а сама задача однозначно разрешима.

В следующей теореме собраны основные критерии однозначной разрешимости СЛАУ.

Лемма 1. Следующие утверждения эквивалентны:
1) решение системы $Ax = b$ существует и единственно при любом b;
2) определитель A отличен от нуля (т.е. $\det(A) \neq 0$).[1)]
3) однородная система $Ax = 0$ имеет лишь решение $x = 0$;
4) система $Ax = b$ имеет единственное решение;
5) матрица A обратима (т.е. существует матрица A^{-1});
6) $\operatorname{rank}(A) = n$.

[1)] Для определителя матрицы A будет использоваться также обозначение $|A|$.

2. О методе Крамера. Решение системы (1) может быть выписано по формулам Крамера

$$x_i = \frac{\Delta_i}{\Delta}, \quad i = 1 : n,$$

где $\Delta = \det(A)$, а Δ_i — определитель, получающийся из определителя матрицы A заменой i-того столбца столбцом свободных членов b.

Казалось бы, формулы Крамера полностью решают задачу построения решения системы линейных уравнений. Однако на практике они не используются. Это объясняется следующим. Напомним, что

$$\det(A) = \sum_{(\alpha_1,\alpha_2,\ldots,\alpha_n)} \pm\, a_{1\alpha_1} a_{2\alpha_2} \cdots a_{n\alpha_n},$$

где $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ — перестановка чисел $(1\ 2\ \ldots\ n)$. Число слагаемых в сумме равно $n!$, поэтому непосредственное вычисление определителя требует более $n!$ арифметических операций (короче — flop: floating point operation), что уже при $n = 30$ недоступно даже для самых мощных ЭВМ. Поэтому для решения систем уравнений применяют другие, более экономичные методы.

3. Типы матриц и методов. Для большинства вычислительных задач, встречающихся на практике, характерным является *большой порядок* n матрицы A, а также *серийность* задачи: требуется решить не одну, а целую серию СЛАУ с одной и той же или близкими матрицами и с разными правыми частями. В связи с этим, там где это возможно, мы будем указывать оценки трудоемкости описываемых методов. Они имеют важное значение для сравнительного анализа численных методов решения задач линейной алгебры.

Вопрос: каких значений может достигать величина n? С системами уравнений какой размерности приходится иметь дело в приложениях на сегодняшний день?

Ответ: величина n может достигать значений $10^6 \sim 10^7$ и есть большая необходимость в решении систем большего порядка.

Системы такой размерности трудно себе представить! Еще труднее себе представить как такие системы решаются. В самом деле, если хранить матрицу A в памяти ЭВМ, то на это потребуется $Q = 8n^2$ байт памяти, если для хранения одного числа отводится 8 байт (как

для чисел типа *double*). При $n = 10^6$ получаем $Q = 8 \cdot 10^{12}$ байт или $Q = 8$ Терабайт. Можно предположить, что для решения СЛАУ такой размерности нужны суперкомпьютеры. Однако, это не так по нескольким причинам. Укажем их.

i) Заполненность матриц. В приложениях приходится иметь дело с двумя типами матриц: с *плотными* и *разреженными* матрицами.

Матрицы, наличием нулевых элементов в которых можно пренебречь, называют *плотными матрицами*. Они хранятся в ЭВМ в виде двумерного массива и обращение к элементу a_{ij} этой матрицы требует небольших накладных расходов.

Матрицы, содержащие относительно небольшое число ненулевых элементов называют *разреженными матрицами*. Хорошим примером такой матрицы является матрица достаточно большого размера n (например, $n \approx 10^6$), на каждой строке которой имеется лишь небольшое число m ненулевых элементов (например, $m \approx 10 \sim 100$). Такие матрицы хранятся в памяти ЭВМ в специальном формате, причем хранятся только ненулевые элементы. Например, для их хранения при $n = 10^6$, $m \approx 10$ требуется ≈ 100 Мегабайт памяти. Обращение к произвольному элементу a_{ij} такой матрицы требует большѝх накладных расходов, но, в зависимости от формата хранения, требует небольших расходов для доступа ко всем ненулевым элементам строки или столбца.

В приложениях плотных матриц порядки $n \approx 10^4$ считаются большими, а $n \approx 10^5$ — сверхбольшими. Для разреженных матриц сверхбольшими считаются порядки $n \approx 10^6 \sim 10^7$.[1)]

ii) Типы численных методов. Методы решения СЛАУ делятся на прямые и итерационные.

Метод называется *прямым*, если для нахождения решения требуется конечное число арифметических операций (+,-,*,/) и извлечений квадратного корня. Например, метод Крамера является прямым методом :-)). Прямые методы требуют хранения матрицы A, а также некоторый объем накладной памяти.

Итерационные методы позволяют за конечное число операций отыскать лишь приближенное решение (с заданной точностью). Они

[1)]Надо понимать, что эти градации являются относительными. Все зависит от конкретной задачи, длины серии, доступного компьютера и программного обеспечения и т.д.

реализуются, чаше всего, как одношаговые или двухшаговые рекуррентные формулы и генерируют последовательность векторов-приближений, сходящуюся к решению. Итерационные методы не требуют обязательного хранения матрицы A; требуется уметь вычислять лишь произведение A на заданный вектор (не редки случаи, когда это можно сделать без хранения A).

iii) Точность решения. Необходимо отметить, что прямые методы только теоретически позволяют найти точное решение задачи, поскольку числа в ЭВМ представляются приближенно (с конечным числом разрядов). Для чисел типа double относительная точность представления равна $2.2 \cdot 10^{-16}$ (т.е. числа типа double имеют около 16 верных значащих десятичных цифр).

Кроме того, для плотных матриц прямые методы требуют порядка $O(n^3)$ flop, каждая из которых также приводит к появлению погрешности в вычислениях. Из-за большого числа операций это приводит к некоторому, а иногда и заметному, накоплению погрешностей округления. Таким образом, в практических вычислениях прямые методы также позволяют найти решение СЛАУ лишь приближенно, хотя, как правило, с высокой точностью (это зависит от свойств матрицы). Отметим, что в практических ситуациях не всегда нужна высокая точность решения.

Итерационные методы оказываются выгодными, если: а) нужна невысокая точность решения; б) при решении дольших и сверхбольших систем; с) при решении СЛАУ со специальными матрицами.

Из-за ограниченности времени и трудности проблемы, вопросов накопления погрешности в вычислениях далее мы касаться не будем. В определенных ситуациях ограничимся лишь замечаниями.

Будем считать далее, что все вычисления осуществляются в точной арифметике.

§ 1. Трудоемкость базовых операций линейной алгебры.

Рассмотрим предварительно трудоемкость некоторых операций.

1. Вычисление суммы векторов. Пусть требуется вычислить сумму $z = x + y$ двух векторов x и y размера n. По определению

$$z_i = x_i + y_i, \quad i = 1 : n.$$

Ясно, что трудоемкость метода составляет n flop.

2. Вычисление скалярного произведения. Трудоемкость вычисления скалярного произведения $(x, y) = \sum_{i=1}^{n} x_i\, y_i$ векторов x и y составляет $2n$ flop (n умножений и n сложений).

3. Вычисление произведения матрицы и вектора. Пусть заданы матрица A размера n и вектор x. Рассмотрим задачу вычисления вектора $b = Ax$. По определению

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

или короче,

$$b_i = \sum_{j=1}^{n} a_{ij}\, x_j, \quad i = 1 : n. \tag{2}$$

Будем говорить, что формула (2) определяет *метод* умножения матрицы A на заданный вектор. Заметим, что он ориентирован на работу со строками матрицы и определяет b_i как скалярное произведение i-той строки A на вектор-строку x.

Непосредственная реализация формул (2) в MATLAB приводит к следующей функции:

```
function b = Axrow(A,x)
n = numel(x);
b = zeros(size(x));
for i=1:n
  for j=1:n
    b(i) = b(i) + A(i,j)*x(j);
  end
end
```

В этой функции компоненты вектора b вычисляются последовательно друг за другом накоплением. Здесь и далее цикл по i означает цикл по строкам, а цикл по j — цикл по столбцам матрицы. Нетрудно видеть, что трудоемкость этой функции равна $2n^2$ flop.

Алгоритм вычисления, реализованный в функции Axrow, называют строчно-ориентированным: в нем цикл по i предшествует циклу по j и в нем обрабатываются в цикле по j строки матрицы. Поменяв порядок циклов придем к другой реализации формул (2) (столбцово-ориентированной). В нем цикл по j предшествует циклу по i:

```
function b=Axcol(A,x)
n = numel(x);
b = zeros(size(x));
for j=1:n
  for i=1:n
    b(i) = b(i) + A(i,j)*x(j);
  end
end
```

В функции Axcol накоплением вычисляются вклады произведения Ax сразу во все компоненты вектора b и ее трудоемкость также равна $2n^2$ flop. В этой функции непосредственно реализован способ вычисления b, основанный на эквивалентной (2) формуле и ориентированной на столбцы матрицы. Он имеет следующий вид:

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix} x_1 + \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{n2} \end{bmatrix} x_2 + \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{nn} \end{bmatrix} x_n.$$

Вместо языка MATLAB можно взять другой язык программирования (например, СИ, Паскаль, Фортран, ...) и написать аналоги функций Axrow и Axcol на этом языке. Практический интерес представляет ответ на следующий вопрос: *какая из полученных функций будет быстрее*, т. е. будет требовать меньшего времени для выполнения? На первый взгляд время работы функций не должно различаться. Однако это не так. *Ответ на этот важный с практической точки зрения вопрос зависит от языка программирования* и связан, главным образом, со способом хранения матриц (способом адресации элементов матриц). Из-за наличия в современных компьютерах многоуровнего кэша последовательное извлечение из оперативной памяти и сохранение чисел, расположенных в соседних ячейках памяти, производится намного быстрее, чем последовательное выполнение тех же операций с элементами, расположенными в памяти далеко друг от друга. В связи с этим, если элементы матрицы в памяти ЭВМ хранятся по строкам (как, например, в C, Паскаль, Python), то быстрее будет выполняться строчно-ориентированная функция Axrow. И наоборот, если элементы матрицы в памяти ЭВМ хранятся по столбцам (Fortran, MATLAB, OpenGL), то быстрее будет выполняться столбцово-ориентированная функция Axcol.

Будем говорить, что функции Axrow и Axcol реализуют *алгоритм* умножения матрицы A на заданный вектор. Эти функции демонстрируют разницу между методом и алгоритмом. В дальнейшем мы ограничимся указанием лишь метода решения задачи.

4. Вычисление произведения двух матриц. Пусть требуется вычислить произведения $C = AB$ двух заданных матриц размера n. По определению j-й столбец C есть произведение матрицы A и j-го столбца B. Так, если b_j есть j-тый столбец B, то $C = AB = A[b_1, b_2, \ldots, b_n] = [Ab_1, Ab_2, \ldots, Ab_n]$, или

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}, \quad i, j = 1 : n.$$

Вычисление c_{ij} (скалярного произведения i-той строки A на j-тый столбец B) требует $2n$ flop, и нужно вычислить n^2 таких элементов. Поэтому трудоемкость вычисления C равна $2n^3$ flop.

§ 2. Простые системы уравнений

Приведем примеры систем уравнений, которые легко решаются.

1. Системы с диагональной матрицей. Пусть D есть диагональная матрица с ненулевыми элементами d_i на диагонали, т. е. $D = \operatorname{diag}(d_1, d_2, \ldots, d_n)$. Тогда система уравнений $Dx = b$ элементарно решается за n flop, и компоненты вектора x находятся по формулам $x_i = b_i/d_i$, $i = 1 : n$.

2. Системы с треугольной матрицей. Матрица A называется *нижней треугольной* (также левой треугольной), если $a_{ij} = 0$ при всех $j > i$. Аналогично, матрица A называется *верхней треугольной* (также правой треугольной), если $a_{ij} = 0$ при всех $i > j$. Как правило, нижние треугольные матрицы обозначаются буквой L (Lower, Left), а верхние треугольные — буквой U (Upper) или R (Right). Таким образом,

$$L = \begin{bmatrix} l_{11} & 0 & \ldots & 0 \\ l_{21} & l_{22} & \ldots & 0 \\ \vdots & \vdots & & \vdots \\ l_{n1} & l_{n2} & \ldots & l_{nn} \end{bmatrix}, \quad U = \begin{bmatrix} u_{11} & u_{12} & \ldots & u_{1n} \\ 0 & u_{22} & \ldots & u_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & u_{nn} \end{bmatrix}.$$

Матрица называется треугольной, если она является либо нижней треугольной, либо верхней треугольной.

Поскольку определитель L равен $\det(L) = l_{11}l_{22}\cdots l_{nn}$, и аналогично $\det(U) = u_{11}u_{22}\cdots u_{nn}$, то треугольные матрицы невырождены тогда и только тогда, когда все их диагональные элементы отличны от нуля.

Система уравнений $Lx = b$ в индексной форме имеет вид

$$\begin{aligned} &l_{11}x_1 && = b_1,\\ &l_{21}x_1 + l_{22}x_2 && = b_2,\\ &\ldots\ldots\ldots\ldots && \ldots\\ &l_{n1}x_1 + l_{n2}x_2 + \ldots + l_{nn}x_n && = b_n. \end{aligned}$$

Решение этой системы находится последовательно: из первого уравнения определяется $x_1 = b_1/l_{11}$, из второго $x_2 = (b_2 - l_{21}x_1)/l_{22}$ и т. д. Таким образом,

$$x_i = \Big(b_i - \sum_{j=1}^{i-1} l_{ij}\, x_j\Big)/l_{ii}, \quad i = 1:n. \tag{3}$$

При $i = 1$ в (3) возникает сумма $\sum\limits_{j=1}^{0} \big(\ldots\big)$. Подобные суммы здесь и далее считаются равными нулю.

Метод (3) решения системы $Lx = b$ называется *прямой подстановкой*. Определим трудоемкость этого метода: при фиксированном i требуется $2(i-1)$ flop для вычисления суммы и дополнительно 2 flop для вычисления x_i. Общее число операций равно

$$Q = 1 + 2\sum_{i=2}^{n} i = n^2 + n - 1 = n^2 + O(n)\ \text{flop}\,,$$

т.к. $1 + 2 + \ldots + m = m(m+1)/2$. Аналогично решается система $Ux = b$. Отличие в том, что сначала определяется $x_n = b_n/u_{nn}$, затем $x_{n-1} = (b_{n-1} - u_{n-1,n}x_n)/u_{n-1,n-1}$ и т. д. Таким образом,

$$x_i = \Big(b_i - \sum_{j=i+1}^{n} u_{ij}\, x_j\Big)/u_{ii}, \quad i = n, n-1, \ldots, 1. \tag{4}$$

Метод (4) решения системы $Ux = b$ называется *обратной подстановкой*. Его трудоемкость также равна $n^2 + O(n)$ flop.

Обратим внимание, что суммарная трудоемкость прямого и обратного хода, т. е. трудоемкость последовательного решения двух треугольных систем $Ly = b$ и $Ux = y$ равна $2n^2 + O(n)$ flop, и при больших значениях n примерно равна трудоемкости вычисления $b = Ax$ при заданном x.

Отметим также замкнутость множества $\mathcal{L}$ всех нижних треугольных матриц (множества $\mathcal{U}$ — всех верхних треугольных матриц) относительно операций сложения и умножения. В самом деле, пусть $L, L_1, L_2 \in \mathcal{L}$. Тогда $L_1 + L_2 \in \mathcal{L}$ (что очевидно), $L_1 L_2 \in \mathcal{L}$ (непосредственно проверяется) и, если L — обратим, то $L^{-1} \in \mathcal{L}$ (см. ниже задачу 6). По определению нулевая матрица и единичная матрица являются элементами как $\mathcal{L}$, так и $\mathcal{U}$. Кроме того, $L_1 + L_2 = L_2 + L_1$, но, вообще говоря, $L_1 L_2 \neq L_2 L_1$. Квадратная матрица

$$L_k = \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & l_{k,k} & 0 & \cdots & 0 \\ 0 & \cdots & l_{k+1,k} & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & l_{n,k} & 0 & \cdots & 1 \end{bmatrix} \tag{5}$$

называется *элементарной нижней треугольной*; она отличается от единичной матрицы лишь элементами k-го столбца. Важное свойство этой матрицы отмечено далее в задаче 5.

3. Системы с ортогональной матрицей. Вещественная матрица Q называется ортогональной, если $Q^T Q = QQ^T = I$, где T означает знак транспонирования, I — единичная матрица. Равенство $Q^T Q = E$ ($QQ^T = E$) согласно правилу умножения матриц означает, что столбцы (строки) Q образуют ортонормированную систему из n векторов. По определению ортогональной матрицы $Q^{-1} = Q^T$. [1)]

Пусть требуется решить систему $Qx = b$. Умножая обе части этого равенства на Q^T, получим $x = Q^T b$. Трудоемкость такого метода решения есть $2n^2$ flop, если Q есть плотная матрица.

Простейшим примером ортогональной матрицы является элементарная матрица перестановок (транспозиция). Матрица P_{ik} называется *элементарной матрицей перестановок*, если она получена из

[1)] Напомним, что B называется обратной к A, если $AB = BA$. Она обозначается через A^{-1}.

единичной матрицы перестановкой строк с номерами i и k. Например, матрицами перестановок третьего порядка являются матрицы

$$P_{12} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad P_{13} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad P_{23} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Далее мы встретимся еще двумя типами ортогональных матриц Q, таких, что $Q = Q^T = Q^{-1}$. Это матрицы отражения и вращения. Матрицы вращения, как и элементарная матрица перестановок являются разреженными (на каждой строке не более двух ненулевых элементов).

Имеется большое количество прямых методов решения СЛАУ. Далее мы ограничимся рассмотрением лишь двух семейств методов, основанных на идее треугольной и ортогональной факторизации матриц: вариантов метода Гаусса и QR разложения.

Задания для самостоятельной работы

ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. Что в линейной алгебре понимают под: a) вектором-столбцом? b) вектором-строкой? c) матрицей?

2. Что понимается под решением СЛАУ?

3. Приведите критерии разрешимости СЛАУ?

4. Укажите формулы Крамера решения СЛАУ.

5. Какие матрицы называются: а) плотными? б) разреженными?

6. Укажите формулы и трудоемкость следующих операций: а) суммы векторов; б) произведения матрицы и вектора; с) произведения двух матриц.

7. Какая матрица называется: а) диагональной; б) нижней треугольной; с) верхней треугольной; д) ортогональной;

8. Укажите формулы решения системы: а) с нижней треугольной матрицей; б) с верхней треугольной матрицей; с) с ортогональной матрицей.

9. Укажите трудоемкость решения системы: а) с нижней треугольной матрицей; б) с верхней треугольной матрицей; с) с ортогональной матрицей.

10. В чем различие между методом и алгоритмом? Приведите примеры.

11. Дайте определение элементарной нижней треугольной матрицы.

12. Дайте определение матрицы перестановок.

13. Какие свойства матрицы перестановок можно отметить?

ЗАДАЧИ И УПРАЖНЕНИЯ

1. Пусть P_{ik} — матрица перестановки. Показать, что вектор $P_{ik}x$ получается из вектора x перестановкой элементов с номерами i, k.

2. Как следствие показать, что матрица $P_{ik}A$ получается из матрицы A перестановкой строк с номерами i, k.

3. Пусть P_{ik} — матрица перестановки. Показать, что $P_{ik}^{-1} = P_{ik}^T = P_{ik}$.

4. Показать, что нижняя треугольная матрица L (с элементами l_{ij}) равна произведению элементарных нижних треугольных матриц L_k (см. (5)), т. е. $L = L_1 L_2 \cdots L_{n-1} L_n$.

Указание. Проведите вычисления в соответствии со следующей расстановкой скобок: $L = L_1(L_2 \cdots (L_{n-2}(L_{n-1}L_n)\cdots)$, т. е. сначала перемножьте $L_{n-1}L_n$, результат умножьте слева на L_{n-2} и т. д.

5. Пусть L_k есть элементарная нижняя треугольная матрица и $l_{kk} \neq 0$. Показать, что

$$L_k^{-1} = \begin{bmatrix} 1 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 1/l_{k,k} & 0 & \dots & 0 \\ 0 & \dots & -l_{k+1,k}/l_{k,k} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & -l_{n,k}/l_{k,k} & 0 & \dots & 1 \end{bmatrix}.$$

6. Пусть L — нижняя треугольная матрица, у которой все элементы главной диагонали отличны от нуля. Показать, что матрица L^{-1} существует и является нижней треугольной матрицей. Показать, что аналогичное верно и для верхней треугольной матрицы.

7. Доказать, что

a) произведение вектор-столбца x на вектор-строку y есть матрица ранга 1.

b) произведение ортогональных матриц есть ортогональная матрица.

c) если матрица A ортогональна, то ортогональными будут и транспонированная к ней и обратная к A матрицы.

d) произведение матриц перестановок есть также матрица перестановок.

§ 3. Метод исключения Гаусса

В основе метода Гаусса, как, впрочем, и многих других методов решения систем линейных алгебраических уравнений

$$Ax = b, \tag{6}$$

лежит следующее утверждение. Пусть матрица B невырождена. Тогда система уравнений

$$BAx = Bb \tag{7}$$

эквивалентна системе (6), т. е. решение системы (7) — решение системы (6) и, наоборот, решение системы (6) — решение системы (7).

Действительно, пусть x — решение системы (7). Тогда

$$B(Ax - b) = 0,$$

но матрица B невырождена, следовательно, $Ax - b = 0$. Обратное утверждение очевидно.

Матрица B выбирается так, чтобы матрица BA была проще матрицы A и решение системы (7) находилось легче, чем решение системы (6). В методе Гаусса матрица B конструируется при помощи элементарных нижних треугольных матриц так, чтобы матрица BA была верхней треугольной. Тогда решение системы (7) становится тривиальной задачей.

Приведем традиционное описание этого метода, ориентированное на операции со строками A.

1. Расчетные формулы. Для удобства изложения положим $A^{(1)} = A$, $b^{(1)} = b$ и запишем исходную систему в индексной форме:

$$\begin{aligned} a_{11}^{(1)} x_1 + a_{12}^{(1)} x_2 + \ldots + a_{1n}^{(1)} x_n &= b_1^{(1)}, \\ a_{21}^{(1)} x_1 + a_{22}^{(1)} x_2 + \ldots + a_{2n}^{(1)} x_n &= b_2^{(1)}, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \\ a_{n1}^{(1)} x_1 + a_{n2}^{(1)} x_2 + \ldots + a_{nn}^{(1)} x_n &= b_n^{(1)}. \end{aligned} \tag{8}$$

Предположим, что $a_{11}^{(1)} \neq 0$ и введем *множители*

$$l_{i1} = a_{i1}^{(1)} / a_{11}^{(1)}, \quad i = 2 : n.$$

Для каждого $i = 2 : n$, умножим обе части первого уравнения в (8) на l_{i1} и вычтем полученное равенство из i-го уравнения. Придем к новой (эквивалентной) системе $A^{(2)} x = b^{(2)}$ вида

$$\begin{aligned} a_{11}^{(1)} x_1 + a_{12}^{(1)} x_2 + \ldots + a_{1n}^{(1)} x_n &= b_1^{(1)}, \\ a_{22}^{(2)} x_2 + \ldots + a_{2n}^{(2)} x_n &= b_2^{(2)}, \\ \cdots\cdots\cdots\cdots\cdots\cdots & \\ a_{n2}^{(2)} x_2 + \ldots + a_{nn}^{(2)} x_n &= b_n^{(2)}. \end{aligned} \tag{9}$$

Согласно описанию, данному выше, новые элементы матрицы и правой части вычисляются по формулам

$$\begin{aligned} a_{ij}^{(2)} &= a_{ij}^{(1)} - l_{i1} a_{1j}^{(1)}, \quad i, j = 2 : n, \\ b_i^{(2)} &= b_i^{(1)} - l_{i1} b_1^{(1)}, \quad i = 2 : n. \end{aligned} \tag{10}$$

Говорят, что в системе (9) неизвестная x_1 исключена из уравнений со второго по n-е или, что матрица системы приведена к верхней треугольной форме в первом столбце. На этом заканчивается описание первого шага метода Гаусса.

На втором шаге проделаем аналогичные вычисления с подсистемой (9), включающей уравнения с номерами $2:n$, и приведем матрицу системы к верхней треугольной форме во втором столбце. Это можно сделать, если $a_{22}^{(2)} \neq 0$. Повторяя вычисления, получим n систем

$$A^{(k)}x = b^{(k)}, \quad k = 1:n,$$

с матрицами вида

$$A^{(k)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1k}^{(1)} & \dots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \dots & a_{2k}^{(2)} & \dots & a_{2n}^{(2)} \\ \vdots & & \ddots & & & \vdots \\ 0 & \dots & 0 & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{bmatrix}, \quad k \geqslant 1.$$

Ясно, что при $k = n$ (после $n-1$ шага) получим систему $A^{(n)}x = b^{(n)}$ с верхней треугольной матрицей

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \dots & a_{2n}^{(2)} \\ \vdots & & \ddots & \vdots \\ 0 & & & a_{nn}^{(n)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_n^{(n)} \end{bmatrix}, \tag{11}$$

решая которую получим искомое решение. Переход от исходной системы (8) к системе (11) называется *прямым ходом метода Гаусса*. Решение системы (11) обратной подстановкой — *обратным ходом*. Элементы $a_{ii}^{(i)}$, $i = 1:n$, называются *ведущими (главными) элементами метода Гаусса* и только на них производится деление в ходе вычислений. Для осуществимости метода *они должны быть отличны от нуля*.

Суммируя сказанное, приходим к следующим расчетным формулам. Для всех $k = 1:n-1$ сначала вычисляются множители

$$l_{ik} = a_{ik}^{(k)}/a_{kk}^{(k)}, \quad i = k+1:n. \tag{12}$$

Затем вычисляются новые элементы матрицы $A^{(k+1)}$ и вектора $b^{(k+1)}$:

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik}a_{kj}^{(k)}, \quad i,j = k+1:n, \tag{13}$$
$$b_i^{(k+1)} = b_i^{(k)} - l_{ik}b_k^{(k)}, \quad i = k+1:n.$$

Можно заметить, что при программной реализации этих формул, элементы $a_{ij}^{(k+1)}$ можно хранить на месте элемента a_{ij} исходной матрицы, также как l_{ik} — на месте элемента a_{ik}, $b_i^{(k+1)}$ — на месте b_i.

2. Трудоемкость метода Гаусса. Ясно, что трудоемкость метода Гаусса вычисляется по формуле

$$Q = \sum_{k=1}^{n-1}(q_{mk} + q_{ak} + q_{bk}) + n^2 + n - 1,$$

где q_{mk}, q_{ak}, q_{bk} есть число операций, необходимых для вычисления множителей на шаге k, новых элементов матрицы $A^{(k+1)}$ и вектора $b^{(k+1)}$ по формулам (12), (13) соответственно, а n^2+n-1 есть трудоемкость обратной подстановки.

Используем хорошо известные формулы:

$$1 + 2 + \cdots + m = \frac{m(m+1)}{2},$$
$$1 + 2^2 + \cdots + m^2 = \frac{m(m+1)(2m+1)}{6}.$$

Ясно, что

$$q_m = \sum_{k=1}^{n-1}(n-k) = \sum_{k=1}^{n-1} k = (n-1)n/2.$$

Для вычисления $b^{(k+1)}$ требуется в два раза больше операций, т. е. $q_b = (n-1)n$. Наконец,

$$q_a = \sum_{k=1}^{n-1} 2(n-k)^2 = 2\sum_{k=1}^{n-1} k^2 = (n-1)n(2n-1)/3.$$

Суммарно, $Q = 2n^3/3 + 3n^2/2 - n/6 - 1 = 2n^3/3 + O(n^2)$ flop.

3. Матричная формулировка метода Гаусса. Для $k = 1 : n-1$ определим элементарную треугольную матрицу L_k, где $l_{i,k}$ есть множители (12) метода Гаусса:

$$L_k = \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & -l_{k+1,k} & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & -l_{n,k} & 0 & \cdots & 1 \end{bmatrix}.$$

Матрица L_k отличается от единичной только поддиагональными элементами k-го столбца. Непосредственными вычислениями легко проверить (убедитесь!), что система уравнений после первого шага метода Гаусса равносильна системе $L_1 Ax = L_1 b$, т. е. $A^{(2)} = L_1 A$, $b^{(2)} = L_1 b$ (см. формулы (10)). Система уравнений после k-го шага равносильна системе $L_k A^{(k)} x = L_k b^{(k)}$, т.е. $A^{(k+1)} = L_k A^{(k)}$, $b^{(k+1)} = L_k b^{(k)}$ (см. формулы (13)). Обозначим $A^{(n)}$ через U. Тогда

$$U = L_{n-1} L_{n-2} \dots L_1 A, \quad b^{(n)} = L_{n-1} L_{n-2} \dots L_1 b.$$

Отсюда находим

$$A = LU,$$

где $L = L_1^{-1} L_2^{-1} \cdots L_{n-1}^{-1}$. Нетрудно видеть (см. далее задачу 5), что

$$L_k^{-1} = \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & l_{k+1,k} & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & l_{n,k} & 0 & \cdots & 1 \end{bmatrix},$$

а матрица L является нижней треугольной с единичной главной диагональю и поддиагональными элементами равными l_{ij} (см. далее задачу 4). Если поддиагональные элементы матрицы L и элементы U хранить на месте соответствующих элементов A, то приходим к следующему алгоритму LU разложения матрицы A (см. расчетные формулы (12), (13)).

```
for k = 1:n-1
  for i = k+1:n
    a(i,k) = a(i,k)/a(k,k);
    for j = k+1:n
      a(i,j) = a(i,j) - a(i,k)*a(k,j);
    end
  end
end
```

Этот алгоритм называется kij – алгоритмом; kji – алгоритм получается перестановкой циклов по i и j.

4. Условия применимости метода Гаусса. Описанный выше метод может быть реализован лишь в том случае, когда все ведущие элементы метода Гаусса отличны от нуля. Для этого невырожденности матрицы недостаточно как показывает следующий пример:

$$A = A^{(1)} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad \det A = -1, \quad A^{(2)} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad a_{22}^{(2)} = 0.$$

Выделим класс матриц, для которых метод Гаусса осуществим. Пусть

$$A_1 = a_{11}, \ A_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}, \ \ldots, A_n = \begin{vmatrix} a_{11} & a_{22} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{vmatrix}$$

есть главные миноры матрицы A.

Теорема 1. *Для того, чтобы все ведущие элементы метода Гаусса были отличны от нуля необходимо и достаточно, чтобы все главные миноры матрицы A были ненулевыми.*

Доказательство. Напомним, что $a_{ij}^{(1)} = a_{ij}$, $i, j = 1 : n$. Пусть $A_i \neq 0$, $i = 1 : n$. Покажем по индукции, что тогда $a_{kk}^{(k)} \neq 0$ для всех $k = 1 : n$. Имеем, $a_{11}^{(1)} = a_{11} = A_1 \neq 0$. Пусть уже доказано, что $a_{11}^{(1)}, a_{22}^{(2)}, \ldots, a_{k-1,k-1}^{(k-1)}$ не равны нулю. Тогда, приводя минор A_k к треугольному виду при помощи преобразований прямого хода метода Гаусса, получим

$$A_k = \begin{vmatrix} a_{11}^{(1)} & a_{22}^{(1)} & \ldots & a_{1k}^{(1)} \\ 0 & a_{22}^{(2)} & \ldots & a_{2k}^{(2)} \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & a_{kk}^{(k)} \end{vmatrix} = a_{11}^{(1)} a_{22}^{(2)} \ldots a_{kk}^{(k)}, \tag{14}$$

следовательно, $a_{kk}^{(k)} \neq 0$, что завершает шаг индукции. Обратное утверждение теоремы есть следствие соотношения (14). $\square$[1)]

Следствием теоремы (1) и разложения $A = LU$ является

Теорема 2. *Пусть все главные миноры матрицы A отличны от нуля. Тогда справедливо единственное представление $A = LU$, где L нижняя треугольная матрица с единичной главной диагональю, U — верхняя треугольная матрица.*

Доказательство. Доказательства требует лишь единственность разложения. Предположим, что имеются два разложения $A = L_1U_1$ и $A = L_2U_2$, т. е. $L_1U_1 = L_2U_2$. Следовательно, $L_2^{-1}L_1 = U_2U_1^{-1}$, причем левая часть этого равенства представляет собой нижнюю треугольную матрицу с единичной диагональю, а правая часть — верхнюю треугольную матрицу. Это возможно только тогда, когда $L_2^{-1}L_1 = I$, $U_2U_1^{-1} = I$, т. е. при $L_1 = L_2$ и $U_1 = U_2$. $\square$

Укажем часто встречающиеся типы матриц, для которых метод Гаусса применим и справедливо разложение $A = LU$.

i) *Симметричные положительно определенные матрицы.* Под $A > 0$ будем понимать, что матрица A положительно определена, т.е.

$$A > 0 \quad \Leftrightarrow \quad (Ax, x) = \sum_{i,j=1}^{n} a_{ij}x_jx_i > 0 \quad \forall\, x \neq 0.$$

В частности, запись $A = A^T > 0$ означает, что A является симметричной положительно определенной матрицей. В соответствии с критерием Сильвестра симметричная матрица положительно определена тогда и только тогда, когда все ее главные миноры положительны.

ii) *Матрицы с диагональным преобладанием.* Матрица, элементы которой удовлетворяют условию:

$$\sum_{j=1,\, j\neq i}^{n} |a_{ij}| < |a_{ii}|, \quad i = 1 : n. \tag{15}$$

называется матрицей с диагональным преобладанием по строкам. Аналогично, матрицы, элементы которой удовлетворяют условию:

$$\sum_{i=1,\, i\neq j}^{n} |a_{ij}| < |a_{jj}|, \quad j = 1 : n.$$

[1)] Значком $\square$ отмечаем конец доказательства.

называются матрицей с диагональным преобладанием по столбцам. Говорят, что A есть матрица диагональным преобладанием, если она имеет диагональное преобладание либо по строкам, либо по столбцам.

Ясно, что если A есть матрица с диагональным преобладанием по строкам, то A^T имеет диагональное преобладание по столбцам.

Теорема 3. *Пусть A есть матрица с диагональным преобладанием. Тогда все ее главные миноры отличны от нуля.*

Доказательство. Достаточно считать, что A имеет диагональное преобладание по строкам. Рассмотрим главный минор A_k, $k \geqslant 1$. Достаточно убедиться, что однородная система линейных уравнений

$$\begin{aligned} &a_{11}x_1 + a_{12}x_2 + \ldots + a_{1k}x_k = 0,\\ &a_{21}x_1 + a_{22}x_2 + \ldots + a_{2k}x_k = 0,\\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ &a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kk}x_k = 0 \end{aligned} \tag{16}$$

с матрицей, составленной из элементов минора A_k, имеет только нулевое решение. Предположим противное и пусть $\max\limits_{1\leqslant j\leqslant k} |x_j| = |x_i|$. Ясно, что $x_i \neq 0$. Поскольку $i \leqslant k$, то из i-е уравнения системы (16) следует

$$a_{ii}x_i = -\sum_{j=1,\, j\neq i}^{k} a_{ij}x_j.$$

Следовательно,

$$|a_{ii}||x_i| \leqslant \sum_{j=1,\, j\neq i}^{k} |a_{ij}|\,|x_j| \leqslant |x_i| \sum_{j=1,\, j\neq i}^{n} |a_{ij}|,$$

т.е. $|a_{ii}| \leqslant \sum_{j=1,\ j\neq i}^{n} |a_{ij}|$, а это противоречит условию (15). $\square$

5. Метод Гаусса с выбором ведущего элемента. Опишем модификацию изученного выше метода Гаусса, который применим для решения систем уравнений с любой невырожденной матрицей.

Выберем среди элементов первого столбца матрицы A максимальный по модулю. Пусть это есть элемент $a_{i_1,1}$. Он не может оказаться равным нулю, так как тогда все элементы первого столбца матрицы A — нули и, значит, $\det(A) = 0$, что противоречит условию $\det(A) \neq 0$.

Умножим обе части уравнения $Ax = b$ на матрицу перестановки $P_{i_1,1}$. В дальнейшем будем обозначать эту матрицу через P_1 (заметим, что она равна единичной, если максимальный по модулю элемент первого столбца матрицы A есть a_{11}). Получим

$$A^{(1)}x = b^{(1)}, \tag{17}$$

где $A^{(1)} = P_1A$, $b^{(1)} = P_1b$. Поясним, что матрица $A^{(1)}$ получается из матрицы A перестановкой первой и i_1-й строк, вектор-столбец $b^{(1)}$ получается из столбца b перестановкой первого и i_1-го элементов. Элементы матрицы $A^{(1)}$ обозначим через $a^{(1)}_{kl}$, элементы столбца $b^{(1)}$ — через $b^{(1)}_k$. По построению $a^{(1)}_{11} \neq 0$.

Теперь можем осуществить первый шаг рассмотренного ранее метода Гаусса и привести матрицу $A^{(1)}$ к верней треугольной форме в первом столбце. Это равносильно умножению обеих частей уравнения (17) на элементарную нижнюю треугольную матрицу L_1. Она была определена при матричной формулировкке метода Гаусса. В результате, придем к системе уравнений

$$A^{(2)}x = b^{(2)}, \tag{18}$$

где $A^{(2)} = L_1A^{(1)} = L_1P_1A$, $b^{(2)} = L_1b^{(1)} = L_1P_1b$. На этом заканчивается первый шаг исключения неизвестных.

На втором шаге среди элементов $a^{(2)}_{22}$, $a^{(2)}_{32}$, $\ldots$, $a^{(2)}_{n2}$ (поддиагональных элементов второго столбца, включая диагональный) найдем максимальный по модулю. Пусть этот элемент есть $a^{(2)}_{i_2,2}$. Он не может равняться нулю. Действительно, если он равен нулю, то все числа $a^{(2)}_{22}$, $a^{(2)}_{32}$, $\ldots$, $a^{(2)}_{n2}$ — нули и тогда, вычисляя $|A^{(2)}|$ разложением по первому столбцу, получим, что $|A^{(2)}| = 0$. С другой стороны, поскольку $|L_1| = 1$, а $|P_1| \neq 0$, то $|A^{(2)}| = |L_1|\,|P_1|\,\det(A) \neq 0$, что приводит к противоречию.

Умножим обе части уравнения (18) на матрицу $P_2 = P_{i_2,2}$, т. е. поменяем местами вторую и i_2-ю строки матрицы $A^{(2)}$. Получим

$$\tilde{A}^{(2)}x = P_2L_1P_1b. \tag{19}$$

По определению элемент $\tilde{a}^{(2)}_{22} \neq 0$. Это позволяет осуществить второй шаг рассмотренного ранее метода Гаусса и привести матрицу $\tilde{A}^{(2)}$ к верней треугольной форме и во втором столбце. Это равносильно

умножению обеих частей уравнения (19) на элементарную нижнюю треугольную матрицу L_2. В результате второго шага получим систему уравнений

$$A^{(3)}x = L_2P_2L_1P_1b,$$

где $A^{(3)} = L_2P_2L_1P_1A$. Продолжая этот процесс, после $n-1$ шага получим систему с верхней треугольной матрицей $U = A^{(n)}$,

$$Ux = f \tag{20}$$

(очевидно, эквивалентную исходной), где

$$U = L_{n-1}P_{n-1}\cdots L_1P_1A, \tag{21}$$

$$f = L_{n-1}P_{n-1}\cdots L_1P_1b.$$

Решение системы (20) не вызывает затруднений.

Замечание 1. Выбор максимального по модулю элемента столбца при выполнении прямого хода метода Гаусса минимизирует влияние ошибок округления. Если не заботиться об ошибках округления, то на очередном шаге прямого хода метода Гаусса можно выбирать любой ненулевой элемент столбца.

Теорема 4. *Пусть* $\det(A) \neq 0$. *Тогда справедливо разложение* $PA = LU$, *где* L *— нижняя треугольная матрица с единичной главной диагональю,* U *— верхняя треугольная матрица,*

$$P = P_{i_{n-1},n-1}P_{i_{n-2},n-2}\cdots P_{i_1,1}$$

— матрица перестановок, $i_k \geqslant k$, $k = 1 : n-1$.

Доказательство. Согласно формуле (21)

$$A = P_1L_1^{-1}\cdots P_{n-2}L_{n-2}^{-1}P_{n-1}L_{n-1}^{-1}U. \tag{22}$$

Здесь мы учли, что произведение P_kP_k есть единичная матрица. Это также позволяет эквивалентно преобразовать (22) к виду

$$P_{n-1}P_{n-2}\cdots P_1A = \Big(P_{n-1}P_{n-2}\cdots P_2L_1^{-1}P_2P_3\cdots P_{n-1}\Big)$$
$$\Big(P_{n-1}\cdots P_3L_2^{-1}P_3\cdots P_{n-1}\Big)\cdots\Big(P_{n-1}L_{n-2}^{-1}P_{n-1}\Big)L_{n-1}^{-1}U =$$
$$= \big(\widetilde{L}_1^{-1}\widetilde{L}_2^{-1}\cdots\widetilde{L}_{n-2}^{-1}L_{n-1}^{-1}\big)U.$$

Отсюда следует утверждение теоремы. Действительно, каждая из матриц $\widetilde{L}_k^{-1}$ представляет собой элементарную нижнюю треугольную матрицу с единичной диагональю, отличающуюся от L_k^{-1} лишь

перестановкой поддиагональных элементов в k-м столбце, а матрица $L = \widetilde{L}_1^{-1} \dots \widetilde{L}_{n-2}^{-1} L_{n-1}^{-1}$ есть нижняя треугольная с единичной диагональю. $\square$

Программная реализации LU разложения матрицы методом Гаусса с выбором ведущего элемента по столбцу осуществляется также, как и описанное ранее LU разложение. Необходимо лишь внести изменения, связанные с перестановкой строк матрицы и запоминанием этих перестановок. Например, kij алгоритм примет вид:

```
function [A,p] = lukij(A)
n = size(A,1);
p = 1:n;
for k = 1:n-1
  [~, I] = max(abs(A(k:n,k)));
  row = I+k-1;
  a([k, row],:) = a([row, k],:);
  p([k, row])   = p([row, k]);
  for i = k+1:n
    a(i,k) = a(i,k)/a(k,k);
    for j = k+1:n
      a(i,j) = a(i,j) - a(i,k)*a(k,j);
    end
  end
end
```

В результате выполнения этой функции, матрицы L и U сохраняются на месте матрицы A. В векторе p сохраняются перестановки строк.

Пусть $[LU, p] = lukij(A)$. Тогда команды $L = tril(LU, -1) + eye(n)$; $U = triu(LU)$ позволяют при необходимости получить L и U. Вектор перестановок p таков, что $A(p, :) = LU$.

Задания для самостоятельной работы

ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. В чем заключается идея метода исключения Гаусса?

2. Приведите расчетные формулы метода исключения Гаусса.

3. Какие элементы метода исключения Гаусса называются ведущими?

4. Применим ли метод Гаусса к системам a) с невырожденной матрицей? b) ортогональной матрицей?

5. Сформулируйте условия применимости метода исключения Гаусса.

6. Укажите трудоемкость метода исключения Гаусса.

7. Пусть система из $n \gg 1$ уравнений решается методом Гаусса за 1 единицу времени. Примерно какое число единиц времени потребует решение системы размера $10n$?

8. Для каких матриц A справедливо разложение $A = LU$?

9. Для каких матриц A справедливо разложение $PA = LU$, где P есть матрица перестановок?

10. В чем отличие метода Гаусса и метода Гаусса с выбором ведущего элемента?

ЗАДАЧИ И УПРАЖНЕНИЯ

1. Пусть диагональные элементы L и U равны единице. Получить формулы для элементов L^{-1} и U^{-1}. Оценить трудоемкость.

2. Пусть M есть множество симметричных матриц размера 2×2 с элементами из отрезка $[0, 1]$, которые можно представить в ЭВМ в плавающей системе double. Определить вероятность того, что СЛАУ со случайно выбранной матрицей $A \in M$ разрешима.

Указание. Число типа double в памяти ЭВМ занимает 64 разряда: 1 бит для знака числа, 11 — для хранения экспоненты и 52 — для мантиссы. Следовательно, числа типа double из интервала [0,1] имеют вид $0.d_1d_2...d_{52}$, где d_k равны 0 или 1, т.е. они образуют равномерную сетку на отрезке [0,1] с шагом $\varepsilon = 2^{-52} \approx 2.2 \cdot 10^{-16}$ (относительная погрешность представления чисел в ЭВМ или машинный ипсилон).

3. Пусть A — кососимметричная вещественная матрица, т.е. $A = -A^T$. Найти те значения α, при которых матрица $A + \alpha I$ обратима (I — единичная матрица) в случае
   a) когда порядок A — нечетное число;
   b) когда порядок A — четное число.

4. Пусть известно LU разложение матрицы A. За сколько арифметических операций можно решить N систем $Ax_k = b_k$, $k = 1 : N$, в этом случае?

Указание. Заметьте, что единичная задача сводится к двум треугольным системам: $Ly = b_k$, $Ux_k = y$.

5. Пусть матрица A симметрична и ее главные миноры отличны от нуля. Докажите, что существует единственное разложение $A = LDL^T$, называемое LDL разложением матрицы A. Здесь L — нижняя треугольная матрица с единичной диагональю, D — диагональная матрица.

6. Пусть матрица A симметричная и ее главные миноры отличны от нуля. Получите расчетные формулы для вычисления элементов L и D в разложении $A = LDL^T$.

Указание. Действуйте также, как и при выводе формул компактного метода Гаусса. В силу симметрии достаточно рассмотреть случаи $i > j$ и $i = j$ (или $i < j$ и $i = j$).

§ 4. Компактные схемы метода Гаусса

Предположим, что LU разложение матрицы A существует. Рассмотрим другие способы его получения, отличные от метода Гаусса.

1. *LU* разложения матрицы. Посмотрим на разложение $A = LU$ как на уравнение $LU = A$ для определения элементов матриц L и U. Тогда получим n^2 уравнений

$$\sum_{k=1}^{n} l_{ik}u_{kj} = a_{ij}, \quad i, j = 1 : n. \tag{23}$$

Поскольку $l_{ii} = 1$, $l_{ik} = 0$, если $k > i$, и $u_{kj} = 0$, если $k > j$, то равенства (23) можно записать в виде

$$\sum_{k=1}^{\min(i,j)} l_{ik}u_{kj} = a_{ij}\,, \quad i,j = 1:n. \tag{24}$$

Рассмотрим два случая: $i > j$ и $i \leqslant j$.

1) При $i > j$ в формуле (24) верхний индекс суммирования равен j. Тогда равенства запишутся в виде

$$\sum_{k=1}^{j-1} l_{ik}u_{kj} + l_{ij}u_{jj} = a_{ij}\,, \quad i > j\,.$$

Отсюда следует, что

$$l_{ij} = \Big(a_{ij} - \sum_{k=1}^{j-1} l_{ik}u_{kj}\Big)/u_{jj}\,, \quad i > j\,. \tag{25}$$

2) При $i \leqslant j$ из (24) получим

$$\sum_{k=1}^{i-1} l_{ik}u_{kj} + u_{ii} = a_{ij}\,, \quad i \leqslant j\,,$$

откуда вытекают формулы

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}\,, \quad i \leqslant j\,. \tag{26}$$

Из формул (25), (26) можно получить различные алгоритмы вычисления элементов L и U, если определить порядок их вычисления.

Например, следующий ijk-алгоритм позволяет вычислять элементы L и U построчно: для всех $i = 1:n$, сначала вычисляются l_{ij} по формулам (25) для всех $j = 1:i-1$, а затем u_{ij} по формулам (26) для всех $j = i:n$ (проверьте!).

В jik-алгоритме элементы L и U вычисляются по столбцам: для всех $j = 1:n$, сначала вычисляются u_{ij} по формулам (26) для всех $i = 1:j$, а затем l_{ij} по формулам (25) для всех $i = j+1:n$ (поверьте!).

Вычисления, аналогичные методу Гаусса, показывают, что трудоемкость этих методов одна и та же и равна $(2/3)\,n^3 + O(n^2)$ flop. Отметим также, что как и в рассмотренном первоначально методе LU разложения, элементы L и U в ходе вычисления можно располагать в соответствующих позициях матрицы A.

2. LDL разложение. Метод Холецкого. Если матрица системы линейных уравнений симметрична и положительно определена, можно добиться двукратного сокращения числа операций и памяти, необходимых для разложения ее на треугольные множители. В основе соответствующего метода лежит

Теорема 5. *Если матрица A симметрична и положительно определена, то справедливы следующие разложения, в которых матрицы определяются однзначно:*

1) $A = LDL^T$, где L — нижняя треугольная матрица с единичной главной диагональю, а D — диагональная матрица с положительными элементами (LDL разложение A);

2) $A = CC^T$, где C — нижняя треугольная матрица с положительной главной диагональю (разложение Холецкого A)[1)].

Доказательство. Поскольку все главные миноры A положительны, то существует единственное треугольное разложение $A = LU$, причем на главной диагонали L стоят единицы, а на главной диагонали U — ведущие элементы метода исключения Гаусса, отличные от нуля.

Пусть $D = \mathrm{diag}(u_{11}, \ldots, u_{nn})$ — диагональная матрица, образованная диагональными элементами U. Определим матрицу $\widetilde{U} = D^{-1}U$, на главной диагонали которой стоят единицы. Тогда $U = D\widetilde{U}$ и $A = LD\widetilde{U}$. Из симметрии матрицы следует, что $\widetilde{U} = L^T$. Следовательно, $A = LDL^T$. Пусть $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)^T$ — орт i-той координатной оси, $e = (L^T)^{-1}e_i$. Тогда

$$0 < (Ae, e) = (LDL^Te, e) = (DL^Te, L^Te) = (De_i, e_i) = u_{ii}.$$

Это доказывает 1). Определим матрицу $D^{1/2} = \mathrm{diag}(u_{11}^{1/2}, \ldots, u_{nn}^{1/2})$. Ясно, что $D^{1/2}D^{1/2} = D$. Тогда $A = LD^{1/2}D^{1/2}L^T = CC^T$, где $C = LD^{1/2}$. Легко видеть, что главные диагонали C и $D^{1/2}$ совпадают. □

Замечание 2. 1) В случае симметричной матрицы в памяти ЭВМ можно хранить только нижнюю треугольную (или верхнюю треугольную) часть A. Это дает практически двукратную экономию памяти ЭВМ, что существенно при больших n.

2) Если известно разложение $A = LDL^T$, то решение системы уравнений $Ax = b$ может быть найдено как последовательное решение трех систем: $Lz = b$, $Dy = z$, $L^Tx = y$. Это потребует $2n^2 + O(n)$ flop (проверьте!)

3) Если известно разложение $A = CC^T$, то решение $Ax = b$ находится как последовательное решение двух треугольных систем: $Cy = b$, $C^Tx = y$. Это потребует также $2n^2 + O(n)$ flop (проверьте!)

[1)] Андре-Луи Холецкий (Andre-Louis Cholesky; 1875-1918) — французский математик

4) Можно сделать предположение, что метод Холецкого в два раза экономичнее метода Гаусса, поскольку вместо двух треугольных матриц L и U в разложении $A = LU$, требуется определить только одну — C. Это же верно и для LDL разложения.

3. LDL разложение. По аналогии с компактной схемой LU разложения получим расчетные формулы для LDL разложения. Будем искать элементы l_{ij} матрицы L и диагональные элементы d_i матрицы D, решая систему (см. упр. 4)

$$LDL^T = A \quad \Leftrightarrow \quad \sum_{k=1}^{n} l_{ik} d_k l_{jk} = a_{ij}\,, \quad i, j = 1 : n.$$

Поскольку $l_{ii} = 1$, $l_{ik} = 0$, если $k > i$, с учетом симметрии получим

$$\sum_{k=1}^{j} l_{ik} d_k l_{jk} = a_{ij}\,, \quad i \geqslant j. \tag{27}$$

1) Рассмотрим случай $i = j$. Тогда из (27) получаем

$$d_j = a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2 d_k. \tag{28}$$

2) При $i > j$ из (27) следует

$$l_{ij} d_j + \sum_{k=1}^{j-1} l_{ik} d_k l_{jk} = a_{ij}.$$

Следовательно,

$$l_{ij} = \Big(a_{ij} - \sum_{k=1}^{j-1} l_{ik} d_k l_{jk}\Big)/d_j, \quad i > j. \tag{29}$$

Эти формулы приводят к следующему методу вычисления элементов L по столбцам: для $j = 1 : n$, сначала вычисляются d_j по формулам (28), а затем l_{ij} по формулам (29) для всех $i = j + 1 : n$ (проверьте!). Аналогично получаются строчно ориентированные формулы (см. упр. 6).

4. Схема внешних произведений метода Холецкого. Используем индукцию по порядку матрицы. Для матрицы первого порядка имеем равенство $a_{11} = \sqrt{a_{11}}\sqrt{a_{11}}$. Пусть утверждение теоремы

верно для матриц порядка $k > 1$. Покажем, что тогда оно верно и для матриц порядка $k+1$. Запишем матрицу A_{k+1} порядка $k+1$ как блочную:

$$A_{k+1} = \begin{bmatrix} A_k & a_k \\ a_k^T & a_{k+1,k+1} \end{bmatrix}. \tag{30}$$

Здесь A_k — матрица порядка k. В силу предположения индукции она симметрична и положительно определена, $A_k = C_k C_k^T$, где C_k — нижняя треугольная матрица с положительными элементами на диагонали. Будем искать разложение A_{k+1} на треугольные множители в виде

$$A_{k+1} = C_{k+1} C_{k+1}^T = \begin{bmatrix} C_k & 0 \\ c_k^T & c_{k+1,k+1} \end{bmatrix} \begin{bmatrix} C_k^T & c_k \\ 0 & c_{k+1,k+1} \end{bmatrix}, \tag{31}$$

где вектор столбец c_k длины $n-1$ и число $c_{k+1,k+1}$ подлежат определению. Выполним умножение в правой части (31), учитывая, что умножение блочных матриц осуществляется по тому же правилу, что и числовых матриц. Получим

$$A_{k+1} = \begin{bmatrix} C_k C_k^T & C_k c_k \\ c_k^T C_k^T & c_k^T c_k + c_{k+1,k+1}^2 \end{bmatrix}. \tag{32}$$

Отметим, что $c_k^T C_k^T = (C_k c_k)^T$. Сравнивая поблочно результат с матрицей A_{k+1} (т.е. формулы (30) и (32)), получим систему линейных уравнений

$$C_k c_k = a_k \tag{33}$$

для определения вектора c_k и уравнение $c_k^T c_k + c_{k+1,k+1}^2 = a_{k+1,k+1}$ для элемента $c_{k+1,k+1}$.

Таким образом, для построения матрицы C, начиная с $c_{11} = \sqrt{a_{11}}$, нужно для всех $k = 2 : n$ решить систему (33) с треугольной матрицей, а затем вычислить $c_{k+1,k+1} = \sqrt{a_{k+1,k+1} - c_k^T c_k}$. Отметим, что $a^T a = (a,a) = |a|^2$ — квадрат длины вектора a.

Этот строчно ориентированный метод вычислений называется схемой внешних произведений. Нетрудно видеть, что его реализация по затратам памяти и объему вычислений действительно оказывается примерно в два раза более экономичной, чем разложение на треугольные множители произвольной невырожденной матрицы (см. упр. 2). В упр. 7 указан другой метод получения расчетных формул.

Задания для самостоятельной работы

ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. В чем заключается идея компактных схем метода Гаусса?

2. Приведите расчетные формулы компактной схемы LU-разложения.

3. Приведите расчетные формулы компактной схемы LDL-разложения.

4. Приведите расчетные формулы метода Холецкого.

5. Пусть разложение $A = LU$ получено методом Гаусса, а разложение $A = L_1U_1$ — компактным методом Гаусса. Совпадают ли матрицы L и L_1, а также U и U_1?

6. Укажите трудоемкость метода Холецкого и сравните ее с трудоемкостью метода Гаусса. Чем объясняется различие?

7. Сравните трудоемкость LDL и LU разложения матрицы. Чем объясняется различие?

ЗАДАЧИ И УПРАЖНЕНИЯ

1. Докажите, что элементы главной диагонали положительно определенной матрицы положительны.

2. Покажите, что трудоемкость метода Холецкого равна $n^3/3 + O(n^2)$ flop.

3. Докажите, что при выполнении условий теоремы 5 нижняя треугольная матрица L в разложении $A = LL^T$ определяется однозначно.

4. Докажите, что из разлжения (31) следует $|A| = |A_k|\, l^2_{k+1,k+1}$.

5. Покажите, что равенство $LDL^T = A$ (см. (27)) равносильно системе

$$\sum_{k=1}^{\min(i,j)} l_{ik} d_k l_{jk} = a_{ij}\,, \quad i \geqslant j.$$

6. Получите строчно ориентированные расчетные формулы LDL разложения.

7. Получите столбцово ориентированные расчетные формулы разложения Холецкого.

Указание. Модифицируйте рассуждения вывода формул LDL разложения матрицы, приведенные выше.

§ 5. Элементарные ортогональные матрицы

Выше мы изучили метод Гаусса и его варианты, эквивалентные разложениям матрицы на треугольные множители. Далее рассмотрим методы, приводящие или основанные на представлении матрицы в виде произведения

$$A = QR, \tag{34}$$

где Q — ортогональная матрица, а R — верхняя треугольная матрица.

Если разложение (34) получено, то решение системы уравнений $Ax = b$ с невырожденной матрицей A сводится к вычислению

вектора $f = Q^T b$ и решению системы уравнений $Rx = f$ с треугольной невырожденной матрицей (проверьте!).

Напомним, что при ортогональном преобразовании длина вектора не меняется, т.к. $|Qx|^2 = (Qx, Qx) = (x, Q^T Qx) = (x, x) = |x|^2$.

При построении разложения (34) используются специальные ортогональные матрицы, позволяющие решить следующую задачу.

Задача 1. *Даны ненулевой вектор* $a \in \mathbb{R}^n$ *и вектор* $e_1 = (1, 0, \ldots, 0)^T$. *Требуется построить ортогональную матрицу* V *такую, что* $Va = \mu\, e_1$, *где* μ — *число (ясно, что* $|\mu| = |a|$*).*

1. Матрицы вращения. Определение 1. *Вещественная матрица* $G = G_{kl} = \{g_{ij}\}_{i,j=1}^n$, $1 \leqslant k < l \leqslant n$, *называется матрицей вращения, если* $g_{ii} = 1$ *при* $i \neq k, l$, $g_{kk} = c$, $g_{ll} = c$, $g_{kl} = -s$, $g_{lk} = s$, *все остальные элементы матрицы* G_{kl} *равны нулю, причем* $|c|^2 + |s|^2 = 1$. *Она имеет вид*

$$G = \begin{pmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & \ddots & & & & & & \\ & & & c & & -s & & & \\ & & & & \ddots & & & & \\ & & & s & & c & & & \\ & & & & & & \ddots & & \\ & & & & & & & 1 & \\ & & & & & & & & 1 \end{pmatrix}.$$

Нетрудно видеть, что G — ортогональная матрица, поскольку ее строки и столбцы образуют ортонормированные системы векторов. Ясно также, что матрицу вращения не надо хранить в ЭВМ в виде квадратной матрицы, достаточно хранить четыре числа (k, l, c, s).

Порождаемое матрицей G преобразование евклидова пространства $\mathbb{R}^n$ в себя представляет собой поворот на угол $\varphi = \operatorname{arctg}(s/c)$ в двумерном подпространстве (плоскости), натянутом на векторы e_k, e_l естественного базиса пространства $\mathbb{R}^n$. Матрица $G^T = G^{-1}$ выполняет поворот в той же плоскости в обратном направлении.

Пусть $a \in \mathbb{R}^n$. Ясно, что $(Ga)_i = a_i$ при $i \neq k, l$, и

$$(Ga)_k = c\,a_k - s\,a_l,$$
$$(Ga)_l = s\,a_k + c\,a_l.$$

Как видим умножение G на заданный вектор a требует 6 flop. Пусть $\rho = (|a_k|^2 + |a_l|^2)^{1/2}$. Положим $c = 1, s = 0$, если $\rho = 0$, и $c = a_k/\rho$, $s = -a_l/\rho$, если $\rho > 0$. Тогда $(Ga)_k = \rho$, $(Ga)_l = 0$.

Следовательно, если $a = (a_1, \ldots, a_n)^T \neq 0$ произвольный вектор, то вектор $a^{(1)} = G_{1n}a$ будет равен $a^{(1)} = (\rho_1, a_2, \ldots, a_{n-1}, 0)^T$, где ρ_1 было определено выше при $k = 1$, $l = n - 1$. Это требует 4 бинарных арифметических операций и одно извлечение корня. Далее, аналогично, $a^{(2)} = G_{1,n-1}a^{(1)} = (\rho_2, a_2, \ldots, a_{n-2}, 0, 0)^T$ также за 4 бинарных арифметических операций и одно извлечение корня. Процесс продолжим, пока не получим вектор $a^{(n-1)} = |a|\,e_1$.

Итак, мы получили следующий **метод решения задачи 1**: если $a \neq 0$ — произвольный вектор из $\mathbb{R}^n$, то за $4(n-1)$ flop и $n-1$ извлечения корня, можно определить пары чисел (c_{n-1}, s_{n-1}), (c_{n-2}, s_{n-2}), $\ldots$, (c_1, s_1) и соответствующие им матрицы вращения $G_{1,n}$, $G_{1,n-1}$, $\ldots$, $G_{1,2}$ такие, что $Ga = |a|\,e_1$, где $G = G_{1,2} \cdots G_{1,n-1}G_{1,n}$ — ортогональная матрица (см. упр. 8). Умножение этого G вектор реализуется за $6(n-1)$ flop.

Таким образом, любой ненулевой вектор при помощи ортогональной матрицы можно преобразовать в вектор, совпадающий по направлению с вектором e_1 естественного базиса.

Замечание 3. Пусть теперь a, b — два произвольных ненулевых вектора пространства $\mathbb{R}^n$. Как только что было показано, существуют ортогональные матрицы $G(a)$ и $G(b)$ такие, что $G(a)a = |a|\,e_1$, $G(b)b = |b|\,e_1$. Отсюда вытекает, что $Ga = \mu\,b$, где $\mu = |a|/|b|$, $G = G^T(b)G(a)$, т.е. для любой пары ненулевых векторов найдется ортогональная матрица, преобразующая первый вектор в вектор, совпадающий по направлению со вторым.

2. Матрицы отражения. Пусть произвольно задан вектор $w = (w_1, w_2, \ldots, w_n\}^T$ единичной длины (матрица $n \times 1$). Матрица

$$H = H(w) = I - 2ww^T = \{\delta_{ij} - 2w_iw_j\}_{i,j=1}^n \tag{35}$$

называется *матрицей отражения*. Отметим ряд ее свойств.

1. Матрица H симметрична (что очевидно) и ортогональна, т.к.

$$H^TH = H^2 = I - 4ww^T + 4w(w^Tw)w^T = I,$$

поскольку $w^T w = |w|^2 = 1$. Таким образом, $H = H^T = H^{-1}$.

2. Пусть $E_{n-1} = \{z \in \mathbb{R}^n : w^T z = (z, w) = 0\}$ — гиперплоскость размерности $n-1$, нормальная к вектору w. Заметим, что

$$Hw = w - 2ww^T w = -w, \quad Hz = z - 2ww^T z = z, \quad z \in E_{n-1}. \quad (36)$$

Следовательно, H имеет однократное собственное значение равное -1, которому соответствует собственный вектор w, и собственное значение $+1$ кратности $n-1$, которому соответствует собственное подпространство E_{n-1}. Отсюда следует, что $\det(H) = -1$ [1)].

3. Пусть x — произвольный вектор, а z его проекция на гиперплоскость E_{n-1}. Ясно, что векторы x, z и w лежат в двумерной плоскости, нормальной к E_{n-1}, и x однозначно представим в виде $x = \alpha w + z$, где α некоторое число. Из равенств (36) вытекает, что $Hx = -\alpha w + z$ (сделайте рисунок!). Можно сказать, таким образом, что отображение, порождаемое матрицей H, выполняет отражение вектора x относительно гиперплоскости E_{n-1}, ортогональной вектору w. Это свойство матрицы H и позволяет называть ее матрицей отражения.

4. Произведение $y = H(w)a$ вычисляется по формуле

$$y = (I - 2ww^T)a = a - \lambda\, w, \quad \lambda = 2\, w^T a,$$

а его трудоемкость равна $4n$ flop (убедитесь в этом!).

5. Пусть заданы векторы a, $e \in \mathbb{R}^n$, $|a| \neq 0$, $|e| = 1$. Рассмотрим задачу построения такой матрицы отражения $H = H(w)$, что $Ha = \mu e$, где $|\mu| = |a|$ (см. замечание 3). Из геометрических соображений ясно, что эта задача имеет два решения[2)]. Положим

$$w = (a - \mu e)/\nu, \quad \nu = |a - \mu e|\,.$$

Имеем

$$\nu^2 = (a - \mu e, a - \mu e) = 2\,(a, a - \mu e)\,, \quad (37)$$

$$H(w)a = a - \frac{2(a, a - \mu e)}{\nu^2}\,(a - \mu e). \quad (38)$$

Из формул (37), (38) следует, что $H(w)a = \mu e$.

[1)] Определитель матрицы равен произведению ее собственных чисел

[2)] Проиллюстрируйте построение вектора w рисунком в двумерном случае.

6. Рассмотрим случай, когда $e = e_1 = (1, 0, \ldots, 0)^T$. Тогда $(a, \mu e) = \mu a_1$. Положим $\mu = \pm|a|a_1/|a_1|$, если $a_1 \neq 0$, иначе примем $\mu = \pm|a|$. Итак, решение задачи $H(w)a = \mu e_1$ определяется формулой (35) при

$$w = \frac{v}{|v|}, \quad v = (a_1 - \mu, a_2, \ldots, a_n)^T, \quad \mu = \pm \begin{cases} |a|, & a_1 = 0, \\ \frac{|a|\, a_1}{|a_1|}, & a_1 \neq 0. \end{cases} \tag{39}$$

Конкретное решение (т. е. знак μ) выбирается из дополнительных соображений, например, с целью получить более устойчивый к погрешностям округления метод при вычислениях на ЭВМ.

Экономное вычисление w по формулам (39) требует $3n$ flop и одно извлечение корня. Матрицу отражения в памяти ЭВМ можно не хранить; достаточно хранить только вектор w.

Итак, мы получили два способа решения задачи 1: на основе матриц вращения и отражения. Оба метода требуют $O(n)$ flop для определения матрицы V, но матрицы отражения оказались более экономичными.

§ 6. QR разложение матриц

Теорема 6. *Пусть A — вещественная квадратная матрица. Тогда существует ортогональная матрица Q такая, что*

$$A = QR,$$

где R — верхняя треугольная матрица.

Доказательство. Доказательство является конструктивным и дает метод построения матриц Q, R, называемый *методом отражения.* Он состоит из $n - 1$ шага и на k-м шаге матрица A преобразуется к матрице, имеющей верхнюю треугольную форму в k-м столбце. Обозначим через I_k единичную матрицу размера k.

Пусть a_j есть j-й столбец A. Если $a_1 = 0$, то перейдем ко второму шагу, полагая $H^{(1)} = I_n$, $A^{(1)} = A$. Иначе, выберем $H^{(1)} = H_1(w_1)$ как такую матрицу отражения, что $H^{(1)}a_1 = \mu_1\, e_1$ и вычислим $A^{(1)} = H^{(1)}A$. По определению

$$A^{(1)} = [H^{(1)}a_1,\ H^{(1)}a_2, \ldots, H^{(1)}a_n]\,.$$

На этом заканчивается первый шаг. Матрица $A^{(1)}$ имеет верхнюю треугольную форму в первом столбце и в блочном виде имеет вид

$$A^{(1)} = \begin{bmatrix} \mu_1 & c_1 \\ 0 & A_1 \end{bmatrix},$$

где $\mu_1 = \pm|a_1|a_{11}/|a_{11}|$, если $a_{11} \neq 0$, в противном случае $\mu_1 = \pm|a_1|$, A_1 — некоторая квадратная матрица размера $n-1$.

Подсчитаем трудоемкость. На вычисление w_1 требуется $3n$ flop. Вычисление произведений $H_1(w_1)a_2, \ldots, H_1(w_1)a_n$ требует $4n(n-1)$ flop. Т.о. трудоемкость первого шага равна $4n^2 - n$ flop, если $a_1 \neq 0$.

Аналогично осуществляется второй шаг с той лишь разницей, что вычисления производятся с матрицей A_1. А именно, если первый столбец A_1 равен нулю, положим $H^{(2)} = I_n$, $A^{(2)} = A^{(1)}$. Иначе, определим $A^{(2)} = H^{(2)}A^{(1)}$, где матрица $H^{(2)}$ имеет вид

$$H^{(2)} = \begin{bmatrix} 1 & 0 \\ 0 & H_2(w_2) \end{bmatrix}.$$

В этом случае

$$A^{(2)} = \begin{bmatrix} 1 & 0 \\ 0 & H_2(w_2) \end{bmatrix} \begin{bmatrix} \mu_1 & c_1 \\ 0 & A_1 \end{bmatrix} = \begin{bmatrix} \mu_1 & c_1 \\ 0 & H_2(w_2)A_1 \end{bmatrix}.$$

Как и на первом шаге, выберем матрицу $H_2(w_2)$ как такую матрицу отражения, что $H_2(w_2)A_1 = \mu_2\bar{e}_1$, где $\bar{e}_1 = (1, 0, \ldots, 0) \in \mathbb{R}^{n-1}$. Размерность этой задачи на единицу меньше, чем на первом шаге, и равна $n-1$. Соответственно, трудоемкость второго шага не превосходит $4(n-1)^2 - (n-1)$ flop. Таким образом, $A^{(2)}$ имеет верхнюю треугольную форму в первых двух столбцах. Легко видеть, что матрица $H^{(2)}$ является ортогональной (см. упр. 5).

Повторяя построения на k шаге определим матрицу $A^{(k)} = H^{(k)}A^{(k-1)}$, где

$$H^{(k)} = \begin{bmatrix} I_{k-1} & 0 \\ 0 & H_k(w_k) \end{bmatrix}, \tag{40}$$

если матрица $A^{(k-1)}$ не имеет верхней треугольной формы в k-м столбце (иначе, полагаем $H^{(k)} = I_n$). Матрица $H_k(w_k)$ размера $n-k+1$ строится как соответствующая матрица отражения.

После $n-1$ шага получим матрицы отражения $H^{(1)}$, …, $H^{(n-1)}$ такие, что $H^{(n-1)}H^{(n-2)}\cdots H^{(1)}A = A^{(n-1)} = R$, где R — верхняя

треугольная матрица с диагональными элементми μ_i. Следовательно, $A = QR$, где $Q = H^{(1)}H^{(2)}\cdots H^{(n-1)}$ суть ортогональная матрица.

Трудоемкость метода равна

$$\sum_{k=2}^{n}\Big(4k^2 - k\Big) = \frac{4}{3}\,n^3 + O(n^2),$$

что при больших значениях n в два раз больше, чем требуется для разложения $PA = LU$ методом Гаусса. $\square$

Важным положительным качеством описанного метода является его применимость для произвольной матрицы без какой-либо перенумерации ее строк, а также его устойчивость к ошибкам округления. Последнее объясняется тем, что при ортогональном преобразовании длина вектора не меняется.

§ 7. Вычисление определителей и обратной матрицы

Факторизация матриц полезна при решении различных задач.

1. Вычисление определителя. Пусть A произвольная невырожденная квадратная матрица размера n. Используя метод Гаусса с выбором ведущего элемента по столбцу за $2n^3/3 + O(n^2)$ flop получим разложение $PA = LU$, где $P = P_{i_{n-1},n-1}P_{i_{n-2},n-2}\cdots P_{i_1,1}$ — матрица перестановок. Тогда

$$\det(P)\,\det(A) = \det(L)\,\det(U) = \det(U) = u_{11}u_{22}\ldots u_{nn}.$$

Поскольку, элементарная матрица перестановок $P_{i_k,k}$ получается из единичной перестановкой строк с номерами k и i_k, то $\det(P_{i_k,k}) = -1$, если на k-том шаге перестановка была, иначе $\det(P_{i_k,k}) = 1$. Таким образом, $\det(P) = (-1)^m$, где m число перестановок строк, совершенных в процессе исключения неизвестных. Окончательно получаем,

$$\det(A) = (-1)^m u_{11}u_{22}\ldots u_{nn}.$$

Аналогичные формулы нетрудно написать и в тех случаях, когда строится разложение матрицы на простые множители.

Надо, однако, иметь в виду, что непосредственное вычисление по этой формуле, как правило, оказываются невозможным: из-за большого числа сомножителей определитель (или результат промежуточных вычислений) зачастую либо слишком велик, либо, наоборот,

слишком мал. Приходится использовать специальные алгоритмы раздельного вычисления мантиссы и порядка определителя.

Примеры: $A = \mathrm{diag}(1:n)$, $A = \mathrm{diag}(1, 1/2, \ldots, 1/n)$.

Заметим, что трудоемкость операций вычисления $\det(A)$ и решения СЛАУ $Ax = b$ имеет один и тот же порядок $2n^3/3 + O(n^2)$ flop.

2. Вычисление обратной матрицы. Задача построения обратной матрицы сводится к решению n систем линейных уравнений с одной и той же матрицей A и различными правыми частями. Действительно, обозначим матрицу A^{-1} через $Y = [y_1, y_2, \ldots, y_n]$, где y_j — столбцы Y. Тогда $AY = I$, где I — единичная матрица. Столбцами I являются единичные орты e_j. Поэтому

$$AY = [Ay_1, Ay_2, \ldots, Ay_n] = [e_1, e_2, \ldots, e_n] = I.$$

Отсюда следуют n равенств

$$Ay_k = e_k, \quad k = 1:n.$$

Рассмотрим два способа вычисления обратной матрицы.

1. Методом Гаусса с выбором ведущего элемента по столбцу вычислим матрицу перестановок P и треугольные матрицы L и U такие, что $PA = LU$. Это потребует $2/3n^3 + O(n^2)$ flop. Тогда получаем $LUy_k = p^k$, где $p^k = P^T e_k$ есть k-й столбец P^T. Нахождение y_k требует решения систем $Ly = p^k$, $Ux^k = y$. Их суммарная трудоемкость равна $2n^2 + O(n)$ flop. Следовательно, матрица A^{-1} этим методом вычисляется за $(2 + 2/3)n^3 + O(n^2) = 8/3n^3 + O(n^2)$ flop.

2. Методом отражения найдем разложение $A = QR$, затратив $4/3n^3 + O(n^2)$ flop. Тогда $RY = Q^T$. Определение Y из этого уравнения потребует $n^3 + O(n^2)$ flop. Суммарно, матрица A^{-1} этим методом вычисляется за $(1 + 4/3)n^3 + O(n^2)$ flop, что на $n^3/3$ flop меньше, чем в первом методе при больших n.

Задания для самостоятельной работы

ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. Дайте определение матрицы вращения. Почему она так назвается?

2. Укажите способ хранения в ЭВМ матрицы вращения.

3. За сколько вращений произвольный вектор можно отразить в вектор, колинеарный данному?

4. Дайте определение матрицы отражения. Почему она так назвается?

5. Укажите способ хранения в ЭВМ матрицы отражения.

6. Сколько арифметических операций требует умножение матрицы отражения на заданный вектор?

7. Чему равен определитель матрицы отражения?

8. Какова трудоемкость метода отражения? Сравните с трудоемкостью метода Гаусса.

9. При больших значениях n выше трудоемкость вычисления определителя матрицы или трудоемкость решения СЛАУ?

10. Каким методом выгоднее вычислять обратную матрицу: на основе LU или QR разложения матрицы?

ЗАДАЧИ И УПРАЖНЕНИЯ

1. Докажите, что произведение матриц отражения есть ортогональная матрица. Найдите определитель произведения матриц отражения.

2. Постройте метод, аналогичный описанному при доказательстве теоремы 6 и основанный на использовании матриц вращения. Оцените трудоемкость.

3. Докажите, что если матрица A невырождена, а диагональные элементы матрицы R считаются положительными, то матрицы Q, R в разложении $A = QR$ определяются однозначно.

4. Укажите метод построения разложений $A = QL$, где Q — ортогональная, L — нижняя треугольная матрицы.

Указание. На первом шаге приведите матрицу к нижнему треугольному виду в последнем столбце и аналогично — на следующих шагах.

5. Докажите, что матрица, определяемая формулой (40) является матрицей отражения и $\det(H^{(k)}) = -1$.

6. Найдите определитель матрицы вращения.

§ 8. Решение разреженных систем уравнений

Кратко рассмотрим вопросы, возникающие при решении СЛАУ $Ax = b$ с разреженной матрицей A. Как упоминалось ранее, хорошим примером такой матрицы является матрица достаточно большого размера n (например, $n \approx 10^5 \sim 10^6$), на каждой строке которой имеется лишь небольшое число m ненулевых элементов (например, $m \approx 10 \sim 100$).

Не каждый метод, пригодный для плотных матриц, является подходящим для решения разреженных СЛАУ большой размерности.

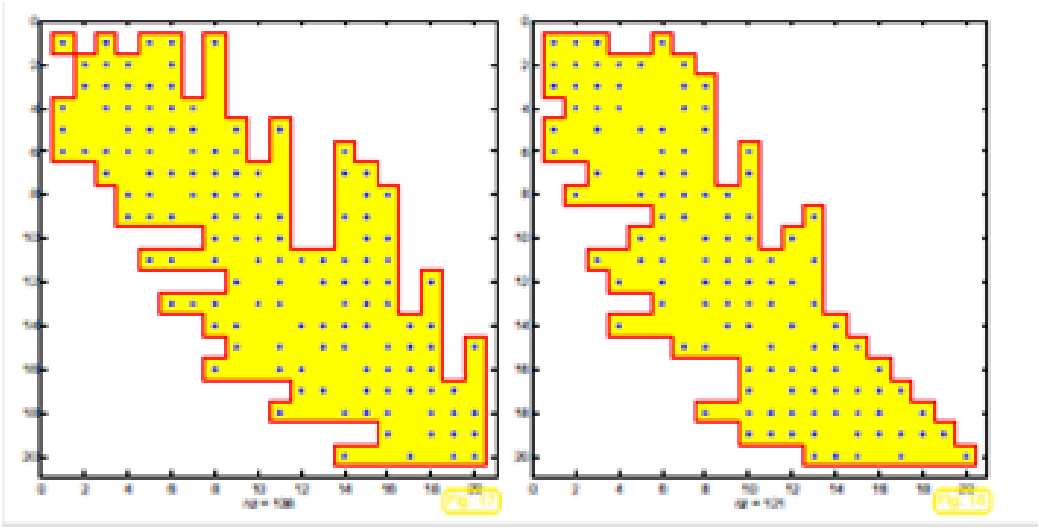

Рис. 1. Портреты и оболочки двух несимметричных матриц.

Интерес представляют только те методы, которые в процессе преобразований исходной матрицы сохраняют ее разреженность. Например, QR метод не является подходящим, т.к. после первых нескольких шагов матрица может стать плотной.

1. О заполнении при LU разложении. Будем предполагать, для упрощения изложения, что существует разложение $A = LU$.

Множество индексов

$$P(A) = \{(i,j) : \ a_{ij} \neq 0, \ i,j = 1:n\}$$

называется портретом матрицы A. Располагая в позициях $P(A)$ матрицы размера $n \times n$ некоторый символ (например, точку), получим графическое изображение ее портрета. На рис. 1 указаны портреты двух несимметричных матриц.

Если $a_{ij} = 0$, но соотвествующий элемент $l_{ij} \neq 0$ или $u_{ij} \neq 0$, то говорят, что в позиции (i,j) произошло заполнение при LU-разложении. Практически важный вопрос заключается в том, насколько велико заполнение и можно ли предсказать заполнение или его отсутствие в определенных позициях?

Для ответа на последний вопрос, определим $\ell_i(A) = \min\{j \leqslant i : \ a_{ij} \neq 0\}$ — номер столбца первого ненулевого элемента на i-той строке. Аналогично определим $m_j(A) = \min\{j \geqslant i : \ a_{ij} \neq 0\}$ — номер строки первого ненулевого элемента в j-том столбце. Следующее множество индексов назовем оболочкой A:

$$E(A) = \{(i,j) : \ell_i(A) \leqslant j \leqslant i = 1:n\} \cup \{(i,j) : m_j(A) \leqslant i \leqslant j = 1:n\}.$$

Элементы A с индексами из оболочки могут быть как ненулевыми, так и нулевыми. Важно, что если $(i,j) \notin E(A)$, то обязательно $a_{ij} = 0$. На рис. 1 цветом выделены оболочки матриц.

Теорема 7. *$E(A) = E(L+U)$, т.е. оболочки матриц A и $L+U$ совпадают. Как следствие заполнение при треугольном разложении может происходить только в позициях из оболочки A.*

Доказательство. Доказательство этой простой, но практически важной теоремы непосредственно следует из рассмотрения расчетных формул LU разложения и вынесено в упражнения. □

Следствие 1. Пусть A — трехдиагональная матрица, т.е. $a_{ij} = 0$, если $|i-j| > 1$ (т.е. ненулевые элементы A могут располагаться только на главной диагонали и двух соседних с ней диагоналях) и $A = LU$. Тогда ненулевые элементы L расположены только на главной диагонали и первой поддиагонали; ненулевые элементы U — только на главной диагонали и первой наддиагонали.

2. Предсказание заполнения. Разреженные матрицы A, L и U хранятся в ЭВМ в специальном формате. Чтобы распределить память под хранение L и U необходимо уметь оценивать заполнение. Для этого используется следующая гипотеза: в формулах типа

$$l_{ij} = \Big(a_{ij} - \sum_{k=1}^{j-1} l_{ik}u_{kj}\Big)/u_{jj}, \tag{41}$$

ненулевые слагаемые взаимно не уничтожаются (т.е., если a_{ij} или хотя бы одно произведение $l_{ik}u_{kj}$ в сумме $\neq 0$, то считается, что $l_{ij} \neq 0$). Это хорошее предположение, т.к. взаимное уничтожение плавающих чисел маловероятно. На основе этой гипотезы получается оценка заполнения. Важно, что при этом анализе можно оперировать только с портретом матрицы, т.е. все операции — целочисленные.

3. О перестановках. Чем меньше заполнение, которое зависит от портрета матрицы, тем эффективнее LU-разложение. В связи с этим рассмотрим СЛАУ $Ax = b$. Если переставить в каком-либо порядке ее строки, то получим СЛАУ с другой матрицей, но с тем же решением x. Ясно, что портреты матриц изменились, изменилось и заполнение. Переставим также столбцы матрицы. Получим систему $B\tilde{x} = \tilde{b}$, причем x и $\tilde{x}$ совпадают с точностью до соотвествующих перестановок компонент. Мы можем выбирать, какую систему решать; очевидно, надо выбрать ту, которая приведет к меньшему заполнению в множителях.

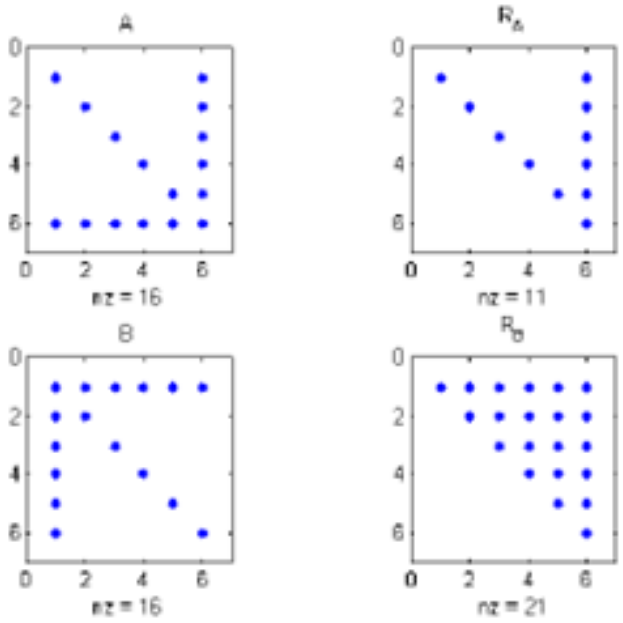

Рис. 2. Матрица с симметричным портретом (слева) и портрет ее множителя U (справа) до перестановки (верхние матрицы) и после (нижние матрицы).

На рисунке 2 сверху приведена матрица A и множитель разложения $R_A = U$. Видно, что заполнения в U не произошло. В то же время, если поменять местами первую и последнюю строки, а так же первый и последний столбцы (для сохранения симметрии при разложении Холецкого), то множитель разложения окажется полностью заполненным (на рис. 2 снизу приведена матрица B и множитель разложения $R_B = U$).

Таким образом, ключевой вопрос заключается в поиске эффективных перестановок строк (и возможно, также столбцов), которые позволят минимизировать заполнение. Как правило, поиск таких эффективных перестановок приводит к NP-полным задачам, поэтому для их поиска применяются эвристические алгоритмы. Они не дают гарантированного оптимального решения, но позволяют существенно уменьшить заполнение в процессе разложения, а значит, сэкономить время и память.

4. Об этапах решения разреженных систем. Разреженные матрицы хранятся в памяти ЭВМ в специальном формате: хранятся ненулевые элементы в виде одномерного массива и информация об их индексах. Современные методы решения СЛАУ включают два этапа.

1) На первом этапе ищутся перестановки строк (и возможно, столбцов), которые позволят уменьшить заполнение в множителях

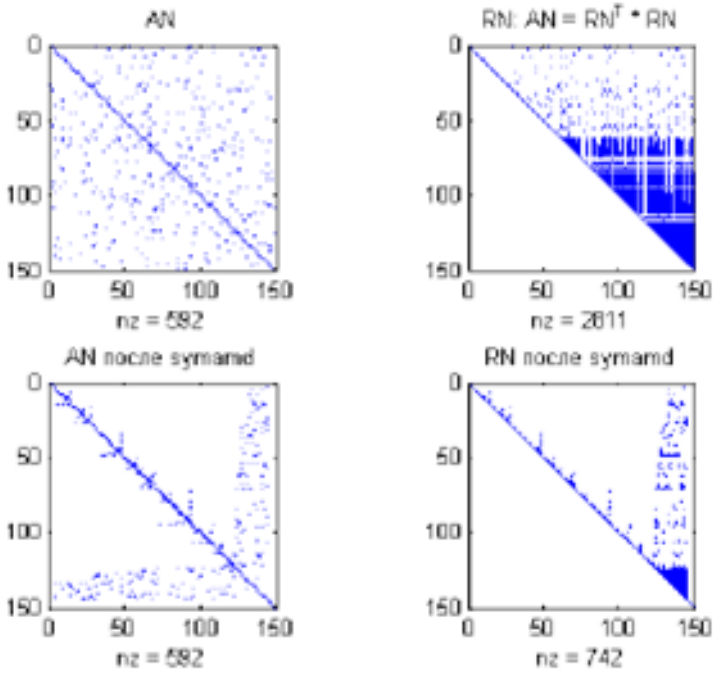

Рис. 3. Демонстрация эффективности алгоритма перенумерации symamd для симметричной матрицы. Наверху: исходная матрица и ее множитель Холецкого (трехкратное заполнение). Внизу матрица после перестановок и ее множитель (незначительное заполнение); nz — число ненулевых элементов матрицы.

разложения. Этот шаг целочисленный, операции производятся только с индексами. На этом шаге выделяется память и формируется портрет множителей (см. рис. (3), демонстрирующий эффективность MatLab-функции перестановок symamd).

2) На втором этапе вычисления производятся в плавающей арифметике и непосредственно вычисляются множители по формулам типа (41) но реализованным с учетом формата хранения матриц.

§ 9. О технологии разреженных матриц

Связанные с разреженными матрицами вопросы типа:

1) форматы хранения разреженных матриц;

2) эффективная реализация базовых операций линейно алгебры над матрицами в этих форматах и векторами (транспонирование, сумма, произведение матриц, произведение матрицы на вектор и т.д.);

3) решением задач линейной (матричной) алгебры (решение СЛАУ, задач на собственные значения и т.д.)

принято относить к технологии разреженных матриц. Кратко рассмотрим форматы хранения разреженных матриц.

1. Координатный формат. Наиболее очевидным способом хранения произвольной разреженной матрицы является координатный формат: хранятся только ненулевые элементы матрицы, и их координаты (номера строк и столбцов). В этом случае требуются три одномерных массива для хранения матрицы A:

1) массив ненулевых элементов матрицы A (обозначим его как a);

2) массив номеров строк матрицы A, соответствующих элементам массива a (обозначим его как i);

3) массив номеров столбцов матрицы A, соответствующих элементам массива a (обозначим его как j).

В качестве примера рассмотрим матрицу

```
A=[1  3  0  1
   0  4  0  2
   2  0  5  0  ];
```

которая может быть представлена в координатном формате как

```
(i,j)  a
(1,1)  1
(3,1)  2
(1,2)  3
(2,2)  4
(3,3)  5
(1,4)  1
(2,4)  2
```

Данный способ представления называют полным, поскольку представлена вся матрица A, и упорядоченным, поскольку ненулевые элементы матрицы перечислены по порядку по столбцам. Через $nnz(A)$ обозначим число ненулевых элементов матрицы A. В данном примере $nnz(A) = 7$.

2. Разреженный строчный формат. Это одна из наиболее широко используемых схем хранения разреженных матриц. Она предъявляет минимальные требования к памяти и в то же время оказывается очень удобной для нескольких важных операций над разреженными матрицами: сложения, умножения, перестановок строк, транспонирования, решения СЛАУ с разреженными матрицами коэффициентов как прямыми, так и итерационными методами и т. д.

В данном формате для хранения матрицы A требуется три одномерных массива:

1) массив ненулевых элементов матрицы A, в котором они перечислены подряд по строкам от первой до последней (обозначим его опять как a);

2) массив номеров столбцов для соответствующих элементов массива a (обозначим его как j);

3) массив указателей позиций в j, с которых начинается описание очередной строки (обозначим его p).

Таким образом, упорядоченные по возрастанию столбцевые индексы ненулевых элементов k-ой строки хранятся в векторе $\ell = p(k) : p(k+1) - 1$, а их значения в векторе $a(\ell) = [a(p(k)), a(p(k)+1), \ldots, a(p(k+1)-1)]$ (в обозначениях MatLab). Если матрица A состоит из n строк, то длина вектора p будет $n+1$, причем $p(n+1) = nnz(A)+1$. Данный способ представления также является полным и упорядоченным, поскольку элементы каждой строки хранятся в соответствии с возрастанием столбцевых индексов. Для нашего примера

```
p = [ 1 4 6 8 ]    j = [ 1 2 4 2 4 1 3 ]    a = [ 1 3 1 4 2 2 5 ]
```

Этот разреженный строчный формат (Compressed Sparse Row или CSR формат) обеспечивает эффективный доступ к строкам матрицы; доступ к столбцам по прежнему затруднен. Поэтому предпочтительно использовать этот способ хранения в тех методах, в которых преобладают строчные операции.

Иногда бывает удобно использовать полный неупорядоченный способ хранения, при котором внутри каждой строки элементы могут храниться в произвольном порядке. Результаты многих матричных операций получаются неупорядоченными, и упорядочивание может быть весьма затратным. В то же время, многие алгоритмы для разреженных матриц не требуют, чтобы представление было упорядоченным.

3. Разреженный столбцевой формат. В этом случае ненулевые элементы матрицы A перечисляются в порядке их появления в столбцах матрицы, а не в строках. Все ненулевые элементы хранятся по столбцам в массиве a; индексы строк ненулевых элементов – в массиве i; элементы массива p указывают на позиции, с которых начинается описание очередного столбца. Для нашего примера

```
i = [ 1 3 1 2 3 1 2 ]    p = [ 1 3 5 6 8 ]    a = [ 1 2 3 4 5 1 2 ]
```

Столбцевые представления могут рассматриваться как строчные представления транспонированных матриц. Разреженный столбцевой формат (CSC формат) обеспечивает эффективный доступ к столбцам матрицы; доступ к строкам затруднен. Поэтому предпочтительно использовать этот способ хранения в тех алгоритмах, в которых преобладают операции над столбцами матрицы

4. Умножение разреженной матрицы на плотный вектор. Пусть известна строчная форма A. Тогда следующая функция реализует требуемую операцию.

```
function y=Ax(p,j,a,x)
%% Ax : Умножение разреженной матрицы A на вектор x.
% p,j,a = строчная форма A в формате CSR

n = numel(p)-1;
y = zeros(n,1);
for k = 1:n
  for l=p(k):p(k+1)-1
    y(k) = y(k) + a(l)*x(j(l));
  end
end
```

§ 10. Метод прогонки

Рассмотрим СЛАУ с трехдиагональной матрицей A. Рассмотрим метод его решения, называемый методом прогонки. Произвольную систему с такой матрицей можно записать в следующем виде:

$$\begin{aligned} b_1x_1 + c_1x_2 &= f_1, \\ a_2x_1 + b_2x_2 + c_2x_3 &= f_2, \\ \dots\dots\dots\dots\dots\dots\dots\dots & \\ a_ix_{i-1} + b_ix_i + c_ix_{i+1} &= f_i, \\ \dots\dots\dots\dots\dots\dots\dots\dots & \\ a_nx_{n-1} + b_nx_n &= f_n. \end{aligned} \tag{42}$$

Разрешим первое уравнение системы относительно x_1. Получим:

$$x_1 = \alpha_2x_2 + \beta_2, \tag{43}$$

где

$$\alpha_2 = -\frac{c_1}{b_1}, \quad \beta_2 = \frac{f_1}{b_1}. \tag{44}$$

Используя соотношение (43) и второе уравнение системы (42), получим аналогичное выражение для x_2. Вообще, если $x_{i-1} = \alpha_i x_i + \beta_i$, то из i-го уравнения системы (42) получим

$$x_i = \alpha_{i+1} x_{i+1} + \beta_{i+1}, \quad i = 1 : n-1, \tag{45}$$

где

$$\alpha_{i+1} = -\frac{c_i}{b_i + a_i \alpha_i}, \quad \beta_{i+1} = \frac{f_i - a_i \beta_i}{b_i + a_i \alpha_i}, \quad i = 2 : n-1. \tag{46}$$

Используя (44) и (46), можно найти все $\alpha_i, \beta_i,\ i = 2 : n$. Записывая теперь соотношение (45) при $i = n-1$ и подставляя результат в последнее уравнение системы (42), получим

$$x_n = (a_n \beta_n - f_n)/(b_n - a_n \alpha_n).$$

Наконец, используя формулы (45) для $i = n-1, n-2, \ldots, 1$, найдем все остальные компоненты вектора x.

Рассмотренный метод есть вариант метода Гаусса, записанный применительно к случаю системы с трехдиагональной матрицей. Процесс вычислений α_i, β_i соответствует прямому ходу метода Гаусса, а вычисления по формулам (45) соответствуют обратному ходу метода Гаусса. Нетрудно подсчитать, что трудоемкость метода равна примерно $8n$ flop.

Метод может быть реализован, когда все знаменатели в формулах (44), (46) отличны от нуля. Учитывая связь метода прогонки с методом Гаусса, можно сказать, что данное условие выполнено, например, когда матрица системы (42) — матрица с диагональным преобладанием, т. е. $|c_1| < |b_1|$, $|a_n| < |b_n|$, $|a_i| + |c_i| < |b_i|$, $i = 2 : n-1$.

Задания для самостоятельной работы

ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. Какие матрицы называются разреженными. Приведите примеры.
2. Дайте определение портрета разреженной матрицы.
3. Дайте определение оболочки разреженной матрицы.
4. Что понимается под заполнением множителей в процессе LU разложения матрицы?
5. Какая имеется связь LU разложения матрицы и ее оболочки?
6. Зависит ли заполнение множителей в процессе LU разложения от престановки строк матрицы? Приведите примеры.

7. Укажите портреты матриц L и U в LU-разложении трехдиагональной матрицы.

8. Опишите основные этапы решения разреженных систем уравнений.

9. Опишите координатный формат хранения разреженной матрицы. Для каких целей используется этот формат?

10. Опишите разреженный строчный формат хранения разреженной матрицы. Для решения каких задач выгоден этот формат?

11. Опишите разреженный столбцевой формат хранения разреженной матрицы. Для решения каких задач выгоден этот формат?

12. Для решения какой задачи применяется метод прогонки? Какова его трудоемкость?

ЗАДАЧИ И УПРАЖНЕНИЯ

1. Докажите теорему 7.

Указание. a) Рассмотрите компактную схему вычисления l_{ij} по формулам ijk-алгоритма: для всех $i = 1 : n$

$$l_{ij} = \Big(a_{ij} - \sum_{k=1}^{j-1} l_{ik}u_{kj}\Big)/u_{jj}, \quad j = 1 : i-1. \tag{47}$$

Последовательно полагая $j = 1 : i-1$, убедитесь, что $l_{ij} = 0$ для всех $j < \ell_i(A)$.
b) Аналогично, используя формулы jik-алгоритма: для всех $j = 1 : n$

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}, \quad i = 1 : j, \tag{48}$$

убедитесь, что $u_{ij} = 0$, если $1 \leqslant i \leqslant m_j(A)$. Отсюда следует утверждение теоремы.

2. Матрицы с ненулевыми элементами на $2m+1$ диагоналях, прилегающих к главной (с индексами $|i-j| \leqslant m$), называются ленточными (с полушириной m).

a) Укажите на рисунке портреты таких матриц и портреты их треугольных множителей.

b) Как такие матрицы можно экономно хранить в ЭВМ?

c) Убедитесь, что трудоемкость их LU разложения равна $O(nm^2)$ flop.

§ 11. Нормы векторов и матриц

Говорят, что на пространстве $\mathbb{R}^n$ введена *норма*, если каждому вектору $x \in \mathbb{R}^n$ однозначно поставлено в соответствие вещественное число $\|x\|$ (читается: норма x). При этом должны быть выполнены следующие условия (*аксиомы нормы*):

1) $\|x\| \geqslant 0$ для $\forall\, x \in \mathbb{R}^n$; равенства $\|x\| = 0$ и $x = 0$ эквивалентны;

2) $\|\alpha x\| = |\alpha|\,\|x\|$ для $\forall\, x \in \mathbb{R}^n$, $\alpha \in \mathbb{R}$;

3) $\|x + y\| \leqslant \|x\| + \|y\|$ для $\forall\, x, y \in \mathbb{R}^n$.

Условие 3) называют *неравенством треугольника*. Отметим, что

4) $\big|\|x\| - \|y\|\big| \leqslant \|x - y\| \quad \forall\, x, y \in \mathbb{R}^n$.

Это неравенство вытекает из аксиомы 3). В самом деле,

$$\|x\| = \|x - y + y\| \leqslant \|x - y\| + \|y\|.$$

Аналогично, $\|y\| \leqslant \|x - y\| + \|x\|$. Неравенство 4) есть просто более краткая запись этих неравенств.

1. Примеры норм на $\mathbb{R}^n$. 1) Пусть $p \geqslant 1$. Равенство $\|x\|_p = \Big(\sum_{k=1}^n |x_k|^p\Big)^{1/p}$ определяет норму. Действительно, аксиомы 1), 2) выполнены очевидным образом, а неравенство 3) при $p = 1$ непосредственно вытекает из свойств модуля, а при $p > 1$ совпадает с известным неравенством Минковского, доказательство которого приведено в конце лекции. Отметим, что случай $p = 2$ соответствует Евклидовой норме вектора, хорошо известной из курса линейной алгебры: $\|x\|_2^2 = |x|^2 = (x, x)$ для любого $x \in \mathbb{R}^n$. Здесь $(\cdot, \cdot)$ — стандартное скалярное произведение на пространстве $\mathbb{R}^n$.

2) Положим $\|x\|_\infty = \max\limits_{1 \leqslant k \leqslant n} |x_k|$. Легко проверяется, что это равенство определяет норму, причем $\|x\|_\infty = \lim_{p \to \infty} \|x\|_p$ (см. упр. 1).

3) Пусть T — произвольная невырожденная матрица, а $\|\cdot\|$ — заданная норма на $\mathbb{R}^n$. Положим $\|x\|_* = \|Tx\|$. Легко видеть, что это равенство определяет новую норму вектора (см. упр. 2).

Определение 2. *Нормы $\|\cdot\|_{(1)}$ и $\|\cdot\|_{(2)}$ эквивалентны, если найдутся положительные постоянные c_1 и c_2 такие, что*[1)]

$$c_1\|x\|_{(1)} \leqslant \|x\|_{(2)} \leqslant c_2\|x\|_{(1)} \quad \forall\, x \in \mathbb{R}^n.$$

Теорема 8. (без доказательства) *Любые две нормы на пространстве $\mathbb{R}^n$ эквивалентны.*

Приведем, например, следующие оценки (см. упр. 3):

$$\|x\|_\infty \leqslant \|x\|_p \leqslant n^{1/p}\,\|x\|_\infty \quad \forall\, x \in \mathbb{R}^n, \quad p \geqslant 1. \tag{49}$$

[1)] Важно иметь в виду, что постоянные c_1, c_2 могут зависеть от n, т. е. от размерности $\mathbb{R}^n$.

2. Норма, порожденная скалярным произведением. Говорят, что на пространстве $\mathbb{R}^n$ введено *скалярное произведение*, если каждой паре векторов $x, y \in \mathbb{R}^n$ однозначно поставлено в соответствие вещественное число (x, y) (читается: скалярное произведение x и y). При этом должны быть выполнены следующие *аксиомы*:

1) $(x, x) \geqslant 0$ для $\forall\, x \in \mathbb{R}^n$; $(x, x) = 0 \quad \rightleftharpoons x = 0$;

2) $(x, y) = (y, x)$ для $\forall\, x, y \in \mathbb{R}^n$;

3) $(\alpha x + \beta y, z) = \alpha(x, z) + \beta(y, z)$ для $\forall\, x, y, z \in \mathbb{R}^n$, $\forall\, \alpha, \beta \in \mathbb{R}$.

Из курса линейной алгебры хорошо известно, что каждое скалярное произведение порождает норму $\|\cdot\|$ по правилу $\|x\| = (x, x)^{1/2}$. Такая норма связана со скалярным произведением неравенством Коши–Буняковского: $|(x, y)| \leqslant \|x\|\, \|y\|$.

Хорошо известный пример скалярного произведения — эвклидово скалярное произведение: $(x, y) = \sum_{i=1}^{n} x_i y_i$. Неравенство Коши–Буняковского в этом случае имеет вид:

$$\Big| \sum_{i=1}^{n} x_i y_i \Big|^2 \leqslant \Big(\sum_{i=1}^{n} x_i^2 \Big) \Big(\sum_{i=1}^{n} y_i^2 \Big).$$

Другой важный пример скалярного произведения — *энергетическое скалярное произведение* $(x, y)_A$. Оно порождается симметричной положительно определенной матрицей A по правилу (см. упр. 4)

$$(x, y)_A = (Ax, y) = \sum_{i,j=1}^{n} a_{ij} x_j y_i.$$

Порождаемая ею норма $\|x\|_A = (x, x)_A^{1/2}$ называется *энергетической нормой* вектора. Согласно неравенству Коши–Буняковского справедлива оценка: $|(x, y)_A| \leqslant \|x\|_A\, \|y\|_A$.

3. Важные числовые неравенства. Напомним, что функция $f(x)$ называется выпуклой на интервале (a, b), если для любых x_1, $x_2 \in (a, b)$ и для $\forall\, \lambda \in [0, 1]$ выполнено неравенство

$$f(\lambda x_1 + (1 - \lambda) x_2) \leqslant \lambda f(x_1) + (1 - \lambda) f(x_2).$$

Геометрически это означает, что любая точка графика функции f на отрезке $[x_1, x_2]$ лежит ниже хорды, стягивающей точки $(x_1, f(x_1))$, $(x_2, f(x_2))$, или на этой же хорде. Например, если функция f непрерывна и дважды непрерывно дифференцируема на (a, b), а ее вторая производная неотрицательна, тогда f — выпуклая функция на (a, b).

Теорема 9. *1) Пусть $a, b > 0$, $p > 1$, $1/p + 1/q = 1$, тогда*

$$ab \leqslant a^p/p + b^q/q \quad \textit{(неравенство Юнга).} \tag{50}$$

2) Неравенство Гельдера. Пусть $a, b \in \mathbb{R}^n$, $p > 1$, $1/p + 1/q = 1$. Тогда

$$\sum_{i=1}^{n} \left|a_i b_i\right| \leqslant \Big(\sum_{i=1}^{n} |a_i|^p\Big)^{1/p} \Big(\sum_{i=1}^{n} |b_i|^q\Big)^{1/q}. \tag{51}$$

3) Неравенство Минковского. Пусть $a, b \in \mathbb{R}^n$, $p > 1$. Тогда

$$\Big(\sum_{i=1}^{n} |a_i + b_i|^p\Big)^{1/p} \leqslant \Big(\sum_{i=1}^{n} |a_i|^p\Big)^{1/p} + \Big(\sum_{i=1}^{n} |b_i|^p\Big)^{1/p}. \tag{52}$$

Доказательство. 1) Легко видеть, что функция $-\ln(x)$ выпукла на интервале $(0, +\infty)$. Поэтому ($\lambda = 1/p$, $(1 - \lambda) = 1/q$))

$$\ln(a^p/p + b^q/q) \geqslant \ln(a^p)/p + \ln(b^q)/q = \ln(ab),$$

что равносильно (50). При $p = q = 2$ неравенство Гельдера называют также неравенством Коши–Буняковского.

2) Неравенство (51) выполнено, если хотя бы один из векторов a, b равен нулю. Иначе, используя неравенство Юнга, будем иметь:

$$\frac{|a_i|}{\left(\sum_{i=1}^n |a_i|^p\right)^{1/p}} \, \frac{|b_i|}{\left(\sum_{i=1}^n |b_i|^q\right)^{1/q}} \leqslant \frac{|a_i|^p}{p\sum_{i=1}^n |a_i|^p} + \frac{|b_i|^q}{q\sum_{i=1}^n |b_i|^q}\,. \tag{53}$$

Суммируя все эти неравенства, получим искомую оценку.

3) Будем считать a, b такими, что левая часть неравенства (52) положительна, так как в противном случае неравенство (52) выполняется очевидным образом. Ясно, что

$$\sum_{i=1}^{n} |a_i + b_i|^p = \sum_{i=1}^{n} |a_i + b_i|^{p-1}|a_i + b_i| \leqslant \sum_{i=1}^{n} |a_i + b_i|^{p-1}|a_i| + \sum_{i=1}^{n} |a_i + b_i|^{p-1}|b_i|\,. \tag{54}$$

Оценим правую часть, используя неравенство Гёльдера. Имеем

$$\sum_{i=1}^{n} |a_i + b_i|^{p-1}|a_i| \leqslant \Big(\sum_{i=1}^{n} |a_i + b_i|^{(p-1)q}\Big)^{1/q} \Big(\sum_{i=1}^{n} |a_i|^p\Big)^{1/p}.$$

Учтем здесь, что $(p-1)q = p$. Аналогично оценим второе слагаемое в правой части (54). В результате получим:

$$\sum_{i=1}^{n} |a_i + b_i|^p \leqslant \Big(\sum_{i=1}^{n} |a_i + b_i|^p\Big)^{1/q} \left(\Big(\sum_{i=1}^{n} |a_i|^p\Big)^{1/p} + \Big(\sum_{i=1}^{n} |b_i|^p\Big)^{1/p}\right).$$

Отсюда следует (52), т.к. $1 - 1/q = 1/p$. $\square$

4. Нормы матриц. Обозначим через M_n множество всех матриц размера $n \times n$. Определяя на нем обычным образом операции сложения двух матриц и умножения матрицы на число, превратим M_n в линейное пространство размерности n^2. Введем на нем норму, т. е. поставим в соответствие каждой $A \in M_n$ число $\|A\|$ (матричную норму) так, что для любых матриц $A, B \in M_n$ и чисел $\alpha \in \mathbb{R}$:

1) $\|A\| \geqslant 0$, равенства $\|A\| = 0$ и $A = 0$ эквивалентны;

2) $\|\alpha A\| = |\alpha| \|A\|$;

3) $\|A + B\| \leqslant \|A\| + \|B\|$;

4) $\|AB\| \leqslant \|A\| \|B\|$.

Замечание 4. Если выполнены только аксиомы 1-3, то говорят, что на M_n введена *векторная норма*. Не всякая векторная норма является матричной. Пусть, например,

$$\|A\| = \max_{1 \leqslant i,j \leqslant n} |a_{ij}|. \tag{55}$$

Очевидно, это — векторная норма, но она не является матричной, т.к., если

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \text{ то } AA = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix},$$

$\|A\| = 1$, $\|AA\| = 2$, и неравенство $\|AA\| \leqslant \|A\| \|A\|$ не выполнено.

5.1. Примеры матричных норм.

a) Положим $\|A\|_{l_1} = \sum\limits_{i,j=1}^{n} |a_{ij}|$. Очевидно, три первых аксиомы нормы выполнены. Проверим аксиому 4). По определению имеем

$$\|AB\|_{l_1} = \sum_{i,j=1}^{n} \left| \sum_{k=1}^{n} a_{ik} b_{kj} \right| \leqslant \sum_{i,j,k=1}^{n} |a_{ik}| |b_{kj}| \leqslant$$

$$\leqslant \sum_{i,j,k,m=1}^{n} |a_{ik}| |b_{mj}| = \|A\|_{l_1} \|B\|_{l_1}.$$

b) Положим $\|A\|_E = \left(\sum_{i,j=1}^{n} |a_{ij}|^2 \right)^{1/2}$. Эта норма порождается естественным скалярным произведением на пространстве $\mathbb{R}^{n^2}$, поэтому три первых аксиомы для нее выполняются. Норму $\|A\|_E$ обычно называют *евклидовой* нормой или нормой *Фробениуса*[1)]. Проверим

[1)] Фердинанд Георг Фробениус (Ferdinand Georg Frobenius; 1849 — 1917) — немецкий математик.

аксиому 4), опираясь на неравенство Коши-Буняковского. Имеем

$$\|AB\|_E^2 = \sum_{i,j=1}^{n} \Big| \sum_{k=1}^{n} a_{ik} b_{kj} \Big|^2 \leqslant \sum_{i,j=1}^{n} \sum_{k=1}^{n} |a_{ik}|^2 \sum_{k=1}^{n} |b_{kj}|^2 =$$
$$= \sum_{i,k=1}^{n} |a_{ik}|^2 \sum_{k,j=1}^{n} |b_{kj}|^2 = \|A\|_E^2 \|B\|_E^2.$$

c) Пусть задана норма $\|\cdot\|$ на $\mathbb{R}^n$. Матричную норму

$$\|A\| = \max_{x \in \mathbb{R}^n,\, x \neq 0} \frac{\|Ax\|}{\|x\|} = \max_{x \in \mathbb{R}^n,\, x \neq 0} \left\| A \frac{x}{\|x\|} \right\| = \max_{\|x\|=1} \|Ax\|, \qquad (56)$$

называют *подчиненной* норме векторов $\|\cdot\|$ или *операторной* нормой.

То, что максимум в (56) достигается, оставим без доказательства, а проверку аксиом 1-3) вынесем в упр. 3. Проверим аксиому 4). Первоначально заметим, что

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} \geqslant \frac{\|Ay\|}{\|x\|} \quad \forall\, y \in \mathbb{R}^n.$$

Отсюда следует важное свойство подчиненной нормы:

5) $\|Ax\| \leqslant \|A\|\,\|x\|$ для любых $x \in \mathbb{R}^n$.

Свойство 5) позволяет проверить аксиому 4). Имеем $\|ABx\| = \|A(Bx)\| \leqslant \|A\|\,\|Bx\| \leqslant \|A\|\,\|B\|\,\|x\|$. Поэтому

$$\|AB\| = \max_{x \neq 0} \frac{\|ABx\|}{\|x\|} \leqslant \|A\|\,\|B\|.$$

Ясно, что при любом способе задания нормы на $\mathbb{R}^n$ подчиненная норма единичной матрицы равна единице.

Не всякая матричная норма подчинена какой либо норме векторов. Например, норма Фробениуса не подчинена никакой норме векторов, так как $\|I\|_E = \sqrt{n}$.

5.2. Примеры подчиненных матричных норм.

а) Пусть норма на пространстве $\mathbb{R}^n$ определена равенством $\|x\|_1 = \sum\limits_{k=1}^{n} |x_k|$. Тогда подчиненная норма матрицы есть

$$\|A\|_1 = \max_{x \in \mathbb{R}^n,\, \|x\|_1 = 1} \|Ax\|_1.$$

Нетрудно видеть, что для любого вектора $x \in \mathbb{R}^n$, $\|x\|_1 = 1$,

$$\|Ax\|_1 = \sum_{i=1}^{n} \Big| \sum_{j=1}^{n} a_{ij}x_j \Big| \leqslant \sum_{i=1}^{n}\sum_{j=1}^{n} |a_{ij}||x_j| = \sum_{j=1}^{n} |x_j| \sum_{i=1}^{n} |a_{ij}| \leqslant$$
$$\leqslant \sum_{j=1}^{n} |x_j| \max_{j=1:n} \sum_{i=1}^{n} |a_{ij}| = \max_{j=1:n} \sum_{i=1}^{n} |a_{ij}| = S.$$

Пусть $S = \sum\limits_{i=1}^{n} |a_{ik}|$ и $e_k = (0, \ldots, 0, 1, 0, \ldots, 0)^T$ есть орт k-той координатной оси. Ясно, что $\|e_k\|_1 = 1$, а $\|Ae_k\|_1 = \sum\limits_{i=1}^{n} |a_{ik}| = S$. Таким образом, $\|Ax\|_1 \leqslant S$ для всех x, $\|x\|_1 = 1$, и $\|Ae_k\|_1 = S$, $\|e_k\|_1 = 1$. Поэтому

$$\|A\|_1 = \max_{\|x\|_1=1} \|Ax\|_1 = \max_{j=1:n} \sum_{i=1}^{n} |a_{ij}|.$$

Эту норму $\|A\|_1$ часто называют *столбцовой* нормой матрицы A.

b) Определим норму на $\mathbb{R}^n$ равенством $\|x\|_\infty = \max\limits_{k=1:n} |x_k|$. Тогда для любого $x \in \mathbb{R}^n$ такого, что $\|x\|_\infty = 1$

$$\|Ax\|_\infty = \max_{i=1:n} \Big| \sum_{j=1}^{n} a_{ij}x_j \Big| \leqslant \max_{i=1:n} \sum_{j=1}^{n} |a_{ij}||x_j| \leqslant$$
$$\leqslant \max_{j=1:n} |x_j| \max_{i=1:n} \sum_{j=1}^{n} |a_{ij}| = \max_{i=1:n} \sum_{j=1}^{n} |a_{ij}| = S.$$

Итак, $\|Ax\|_\infty \leqslant S$ при любом x, $\|x\|_\infty = 1$. Докажем, что найдется e такой, что $\|e\|_\infty = 1$ и $\|Ae\|_\infty = S$. Тогда получим, что $\|A\|_\infty = S$.

Пусть $S = \sum\limits_{j=1}^{n} |a_{kj}|$. Определим компоненты e как

$$e_j = \begin{cases} a_{kj}/|a_{kj}|, & a_{kj} \neq 0, \\ 1, & a_{kj} = 0, \end{cases} \qquad j = 1 : n.$$

Ясно, что $\|e\|_\infty = 1$, причем что для любого $i = 1 : n$

$$\Big| \sum_{j=1}^{n} a_{ij}e_j \Big| \leqslant \sum_{j=1}^{n} |a_{ij}| \leqslant \max_{i=1:n} \sum_{j=1}^{n} |a_{ij}| = S, \tag{57}$$

а для $i = k$ по определению e_j получим

$$\Big|\sum_{j=1}^{n} a_{ij}e_j\Big| = \sum_{j=1}^{n} |a_{kj}| = S. \tag{58}$$

Из (57), (58) следует $\|Ae\|_\infty = S$. Таким образом,

$$\|A\|_\infty = \max_{\|x\|_\infty=1} \|Ax\|_\infty = \max_{i=1:n} \sum_{j=1}^{n} |a_{ij}|.$$

Норму $\|A\|_\infty$ часто называют *строчной* нормой матрицы A.

c) Введем матричную норму, подчиненную евклидовой норме вектора $\|x\|_2 = (x,x)^{1/2}$. Для любого $x \in \mathbb{R}^n$, $\|x\|_2 = 1$, справедливо равенство $\|Ax\|_2^2 = (Ax, Ax) = (A^TAx, x)$. Матрица $S = A^TA$ симметрична и положительно определена. Согласно спектральному разложению $S = H^T\Lambda H$, где H — ортогональная матрица, $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ диагональная матрица из собственных чисел S. Поэтому

$$\|Ax\|_2^2 = (H^T\Lambda Hx, x) = (\Lambda Hx, Hx) = (\Lambda y, y) = \sum_{i=1}^{n} \lambda_i y_i^2 \leqslant \max_{i=1:n} \lambda_i, \tag{59}$$

т.к. $y = Hx$, $\|y\|_2 = \|x\|_2 = 1$. С другой стороны, если $\max_{i=1:n} \lambda_i = \lambda_k$, то выбирая в (59) x как решение уравнения $Hx = e_k$, где e_k есть орт k-той оси, получим $\|Ax\|_2^2 = \lambda_k$, что вместе с (59) приводит к равенству

$$\|A\|_2 = \max_{\|x\|_1=1} \|Ax\|_2 = \sqrt{\max_{i=1:n} \lambda_i(A^TA)},$$

где $\lambda_i(A^TA)$ есть собственное число матрицы A^TA.

Отметим следующий важный для многих приложений частный случай симметричной матрицы, когда $A = A^T$. В этом случае, из равенства $Ax = \lambda(A)x$ вытекает $A^2x = \lambda(A)Ax = \lambda^2(A)x$. Поэтому

$$A = A^T \quad \Rightarrow \quad \|A\|_2 = \max_{\|x\|_1=1} \|Ax\|_2 = \max_{i=1:n} |\lambda_i(A)|,$$

где через $\lambda_i(A)$ обозначены собственные числа матрицы A.

Максимальное по модулю собственное число матрицы A принято обозначать через $\rho(A)$ и называть спектральным радиусом матрицы A. Норму $\|A\|_2$ в связи с этим часто называют *спектральной*.

Замечание 5. Вычисление собственных чисел матрицы, вообще говоря, — довольно сложная задача. Поэтому полезно получить некоторую оценку величины $\|A\|_2$, просто выражаемую через элементы матрицы A.

Теорема 10. *Для любой матрицы A справедливо неравенство $\|A\|_2 \leqslant \|A\|_E$.*

Доказательство. Используем стандартное обозначение $\mathrm{tr}(S)$ для следа матрицы S, вычисляемого как сумма элементов ее главной диагонали. Известно, что он равен сумме собственных чисел S. Поэтому $\mathrm{tr}(A^TA) = \sum_{k=1}^n \lambda_k(A^TA) \geqslant \max_{k=1:n} \lambda_k(A^TA) = \|A\|_2^2$. С другой стороны легко вычислить, что $\mathrm{tr}(A^TA) = \sum\limits_{i,j=1}^{n} |a_{ij}|^2$. Следовательно,

$$\|A\|_2 \leqslant \Big(\sum_{i,j=1}^{n} |a_{ij}|^2 \Big)^{1/2} = \|A\|_E. \quad \square$$

§ 12. Устойчивость решений СЛАУ

Когда мы говорим о решении СЛАУ $Ax = b$ в практическом смысле, то должны отдавать себе отчет в том, что данные задачи, а именно, матрица A и вектор правой части b, заданы приближенно. То есть, вместо матрицы A известна матрица $\bar{A} = A + \Delta A$, а вместо вектора b задан вектор $\bar{b} = b + \Delta b$. Матрицу ΔA называют возмущением матрицы A, вектор Δb — возмущением вектора b. Причинами появления возмущений могут служить:

1) ввод чисел в ЭВМ. При этом возмущения имеют относительный порядок $\approx 10^{-16}$ (тип double). Это возмущение всегда присутствует, поскольку мы решаем задачу при помощи ЭВМ;

2) погрешности алгоритмов (если элементы матрицы A и вектора b вычисляются приближенно при помощи некоторого алгоритма). Величины возмущений зависят от точности используемых алгоритмов;

3) погрешности измерений (погрешности приборов, если элементы A и b получаются в результате измерений). Величина возмущений зависит от точности измерений, и т.д.

Таким образом, вместо системы $Ax = b$, на самом деле, мы решаем СЛАУ $\bar{A}\bar{x} = \bar{b}$. Решение этой возмущенной системы $\bar{x}$, конечно, не совпадает с решением исходной системы. Возникает естественный вопрос, на который мы получим некоторый ответ далее:

Если возмущения малы, т.е. данные задач $Ax = b$ и $\bar{A}\bar{x} = \bar{b}$ мало отличаются друг от друга, то будут ли близки их решения?

При ответе на этот вопрос мы будем предполагать, что обе задачи решаются точно, т.е. игнорируются ошибки методов их решения и ошибки округления при выполнении арифметических операций.

Теорема 11. *1) Пусть x и $\bar{x}$ есть решения систем $Ax = b$ и $A\bar{x} = \bar{b}$ соответственно,* $\det(A) \neq 0$. *Тогда*

$$\frac{\|x - \bar{x}\|}{\|x\|} \leqslant \operatorname{cond}(A) \frac{\|b - \bar{b}\|}{\|b\|}.$$

2) Если же $\bar{A}\bar{x} = b$, то

$$\frac{\|x - \bar{x}\|}{\|\bar{x}\|} \leqslant \operatorname{cond}(A) \frac{\|A - \bar{A}\|}{\|A\|}.$$

Здесь число $\operatorname{cond}(A) = \|A^{-1}\| \, \|A\|$ *называется числом обусловленности матрицы A, норма вектора — произвольная, норма матрицы — подчиненная норме вектора (операторная).*

Доказательство. 1) Имеем $A(x - \bar{x}) = b - \bar{b}$, Следовательно,

$$\|x - \bar{x}\| = \|A^{-1}(b - \bar{b})\| \leqslant \|A^{-1}\| \, \|b - \bar{b}\|. \tag{60}$$

С другой стороны, $\|b\| = \|Ax\| \leqslant \|A\| \, \|x\|$. Умножим это неравенство на (60) и поделим обе части полученного равенства на $\|x\| \, \|b\|$. Получим 1). Для доказательства 2) заметим, что

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

Отсюда следует оценка

$$\|\bar{x} - x\| \leqslant \|A^{-1}\| \, \|(A - \bar{A})\| \, \|\bar{x}\|.$$

Поделим обе части этой оценки на $\|\bar{x}\|$ и умножим и поделим правую часть на $\|A\|$. Получим 2). $\square$

В общем случае справедлива аналогичная оценка

Теорема 12. *Пусть x и $\bar{x}$ есть решения систем $Ax = b$ и $\bar{A}\bar{x} = \bar{b}$, соответственно,* $\det(A) \neq 0$, $\triangle A = A - \bar{A}$, $\triangle b = b - \bar{b}$. *Тогда, если* $\|A^{-1}\triangle A\| < 1$, *то*

$$\frac{\|x - \bar{x}\|}{\|x\|} \leqslant \frac{\operatorname{cond}(A)}{1 - \|A^{-1}\triangle A\|} \Big(\frac{\|\triangle b\|}{\|b\|} + \frac{\|\triangle A\|}{\|A\|} \Big).$$

Как видим, оценка возмущения решения прямо пропорционально числу обусловленности матрицы, причем $cond(A) \geqslant 1$. В самом деле, $I = A\,A^{-1}$, т.е. $1 = \|I\| = \|A\,A^{-1}\| \leqslant \|A\|\,\|A^{-1}\| = \mathrm{cond}(A)$. У некоторых матриц $cond(A)$ может быть велико, в этом случае оценка не гарантирует малость возмущения решения. Конечно, мы получили лишь оценки, но на классе всех СЛАУ они являются точными.

Матрицы с очень большим числом обусловленности принято называть *плохо обусловленными*. СЛАУ с плохообусловленными матрицами требуют особого подхода и, как правило, не могут быть удовлетворительно решены рассмотренными нами прямыми методами.

Пример 1. Матрица Гильберта — хороший пример плохо обусловленной матрицы.

$$H_n = \left\{\frac{1}{i+j-1}\right\}_{i,j=1}^{n}. \quad \text{СЛАУ} \quad Ax = b, \quad x = (1,1,\ldots,1)^T.$$

$$\det(H_n) \approx 0.6\,n^{-1/4}\,(2\pi)^n 4^{-n^2}, \quad \mathrm{cond}_2(H_n) = O(2.2^{4n}/\sqrt{n}).$$

n	4	8	10	12	15
cond_2	1.6e+5	1.5e+10	1.60e+13	1.7e+016	2.5e+17
err_2	1.9e-13	1.0e-7	2.7e-4	0.08	1.3

Таблица 1.1. Матрица Гильберта: $\mathrm{cond}_2 = \mathrm{cond}_2(H_n)$ и относительная погрешность решения СЛАУ err_2 в евклидовой норме.

В таб. 1.1 представлены результаты вычисления в MatLab числа обусловленности матрицы Гильберта, а также относительной погрешности решения СЛАУ: ее точное решение $x = (1,1,\ldots,1)^T$, $\bar{x} = H\backslash b$ — решение этой системы, вычисленное в MatLab. Отношение $err_2/\mathrm{cond}_2 \approx 10^{-17}$.

Пример 2. Рассмотрим СЛАУ $Ax = b$, где A — модифицированная Жорданова клетка ($a > 1$):

$$A = \begin{pmatrix} 1 & a & & & \\ & 1 & a & & \\ & & \ddots & & \\ & & & 1 & a \\ & & & & 1 \end{pmatrix}, \qquad A^{-1} = \begin{pmatrix} 1 & -a & a^2 & \cdots & (-1)^{n-1}a^{n-1} \\ & 1 & -a & a^2 & \\ & & \ddots & & a^2 \\ & & & 1 & -a \\ & & & & 1 \end{pmatrix}.$$

У обратной матрицы главная диагональ состоит из 1, наддиагонали соответственно состоят из $-a$, a^2, $-a^3$ и т.д.

Будем считать, что $b = (1+a, 1+a, \ldots, 1+a, a)^T$, так что известно точное решение СЛАУ $x = (1,1,\ldots,1)^T$, $\|x\|_\infty = 1$. Имеем,

$$\|A\|_\infty = 1 + a, \quad \|A^{-1}\|_\infty = 1 + a + a^2 + \ldots + a^{n-1} = \frac{a^n - 1}{a - 1},$$

$$\mathrm{cond}_\infty(A) = (1+a)\,\frac{a^n - 1}{a - 1}.$$

Рассмотрим систему $A\bar{x} = \bar{b}$, где $\bar{b} = b - \varepsilon e_n$, $e_n = (0, 0, \ldots, 0, 1)^T$.
Так как $\|b\|_\infty = 1 + a$, $\|b - \bar{b}\|_\infty = \varepsilon$, то согласно теореме 2 имеем

$$\frac{\|x - \bar{x}\|_\infty}{\|x\|_\infty} \leqslant \mathrm{cond}_\infty(A)\, \frac{\|b - \bar{b}\|_\infty}{\|b\|_\infty} = \frac{a^n - 1}{a - 1}\, \varepsilon \approx a^{n-1}\varepsilon. \tag{61}$$

С другой стороны, пусть $z = x - \bar{x}$. Тогда $Az = \varepsilon\, e_n$. Эта система легко решается обратным ходом. Получаем, $z_n = \varepsilon$, $z_{n-1} = -a\,\varepsilon$, $z_{n-2} = a^2\,\varepsilon$, …, $z_1 = \pm\, a^{n-1}\,\varepsilon$. Таким образом, $\|x - \bar{x}\|_\infty = \|z\|_\infty = a^{n-1}\,\varepsilon$ и

$$\frac{\|x - \bar{x}\|_\infty}{\|x\|_\infty} = a^{n-1}\,\varepsilon. \tag{62}$$

Как видим, теоретическая оценка (61) практически совпадает с точной величиной относительной погрешности (62).

Задания для самостоятельной работы

ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. Дайте определение нормы векторов. Приведите примеры.

2. Дайте определение норм $\|x\|_1$, $\|x\|_2$, $\|x\|_\infty$.

3. Дайте определение эквивалентных норм векторов. Приведите примеры эквивалентных норм.

4. Дайте определение скалярного произведения векторов. Приведите примеры.

5. Дайте определение энергетического скалярного произведения.

6. Дайте определение энергетической нормы вектора. Какой матрицей оно порождается?

7. Что понимают под сходимостью последовательности векторов к некоторому вектору?

8. Что означает сходимость векторов в норме $\|x\|_\infty$?

9. Если последовательность векторов сходится к некоторому вектору в одной норме, сходится ли она в другой норме к тому же вектору?

10. Запишите неравенство Коши–Буняковского в общем случае. Приведите примеры.

11. Дайте определение векторной нормы матриц. Приведите примеры.

12. Дайте определение подчиненной нормы матрицы. Чему равна подчиненная норма единичной матрицы?

13. Дайте определение норм $\|A\|_1$, $\|A\|_2$, $\|A\|_\infty$.

14. Дайте определение нормы $\|A\|_2$ симметричной матрицы.

15. Что понимается под возмущение матрицы или вектора?

162. Дайте определение числа обусловленности матрицы.

17. Приведите оценку возмущения решения СЛАУ при возмущении ее правой части.

18. Приведите оценку возмущения решения СЛАУ при возмущении ее матрицы.

19. Какая матрица называется плохо обусловленной?

20. Приведите примеры плохо обусловленных матриц.

21. Связана ли плохая обусловленность матрицы с малостью ее определителя и как?