БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра вычислительной математики

Б. В. Фалейчик

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ АЛГЕБРЫ: БАЗОВЫЕ ПОНЯТИЯ И АЛГОРИТМЫ

Учебно-методическое пособие

Минск
2010

УДК 519.612
ББК 22.19
Ф19

Утверждено на заседании
кафедры вычислительной математики
15 апреля 2010 г., протокол № 11.

Фалейчик, Б. В.

Ф19 Вычислительные методы алгебры: базовые понятия и алгоритмы : учеб.-метод. пособие / Б. В. Фалейчик. — Минск : БГУ, 2010. — 42 с.

Рассматриваются практические вопросы машинных вычислений, понятие обусловленности вычислительной задачи. Основное внимание уделено точным методам решения систем линейных алгебраических уравнений: методам Гаусса, LU-разложения, а также методам, основанным на ортогональных преобразованиях. Некоторые из содержащихся в приложении задач и упражнений можно использовать в качестве основы для лабораторных работ.

УДК 519.612
ББК 22.19

Содержание

1 Машинная арифметика

1.1 Числа с плавающей точкой

Для того, чтобы использовать вещественные числа в машинных вычислениях, необходимо решить следующую общую проблему: каким образом сохранить произвольное $x \in \mathbb{R}$ в ограниченном количестве ячеек памяти? Существует несколько способов решения этой проблемы, и наиболее распространённым является представление x в виде числа с плавающей точкой.

Пусть $\beta \in \mathbb{N}$ — основание системы счисления, $p \in \mathbb{N}$ — число значащих разрядов, d_i — цифры. Вещественное число вида

$$\pm \underbrace{d_0.d_1 d_2 \dots d_{p-1}}_{m} \times \beta^e, \quad 0 \leqslant d_i < \beta, \tag{1.1}$$

называется *числом с плавающей точкой* (ЧПТ).Число $m \in \mathbb{R}$ называют *мантиссой* или *значащей частью*. Число $e \in \mathbb{Z}$ называют *показателем*, или *экспонентой* (не путать с числом e).

Представление (1.1) для любого x, очевидно, не является единственным — оно зависит от «положения точки»:

$$0.0001234 = 0.0012340 \times 10^{-1} = 1.2340000 \times 10^{-4}.$$

Поэтому по умолчанию используется так называемая нормализованная форма записи ЧПТ, в которой точка ставится после первой значащей цифры.

ЧПТ вида (1.1) с ненулевым первым разрядом ($d_0 \neq 0$) называется *нормализованным*. Множество всех нормализованных ЧПТ с основанием β, p-разрядной мантиссой, и $e_{\min} \leqslant e \leqslant e_{\max}$ условимся обозначать $\mathbb{F}_1(\beta, p, e_{\min}, e_{\max})$ или просто $\mathbb{F}_1$.

$\triangleright_1$ Вычислите мощность множества $\mathbb{F}_1$ в общем случае.

Таблица 1. Примеры ЧПТ в нормализованной форме. В скобках указаны значения в десятичной системе счисления.

β	p	x	m	e
10	6	0.000031415	3.14150	-5
2	10	10001.101 (17.625)	1.000110100	100 (4)
16	5	ABC.D (2748.8125)	A.BCD0	2 (2)

$\triangleright_2$ Запишите в виде нормализованного числа с плавающей точкой следующие числа: 2010.02_{10}, 0.0010101_2, $E2.E4_{16}$.

1.2 Двоичные числа с плавающей точкой

Рассмотрим подробно случай $\beta = 2$, так как именно такая арифметика используется в большинстве современных компьютеров. Возьмём $p = 3$, $e_{\min} = -1$, $e_{\max} = 2$. Все соответствующие положительные ЧПТ приведены в таблице, они же изображены на рис. 1.

Таблица 2. Все положительные элементы множества $\mathbb{F}_1(2, 3, -1, 2)$.

$m \diagdown e$	-1		0		1		10 (2)	
1.00	0.1	(0.5)	1	(1)	10	(2)	100	(4)
1.01	0.101	(0.625)	1.01	(1.25)	10.1	(2.5)	101	(5)
1.10	0.110	(0.75)	1.1	(1.5)	11	(3)	110	(6)
1.11	0.111	(0.875)	1.11	(1.75)	11.1	(3.5)	111	(7)

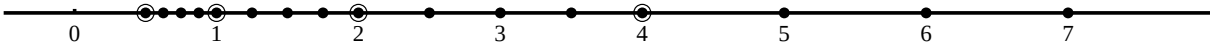


Рисунок 1. Распределение машинных чисел из таблицы 2 на числовой прямой. Степени двойки выделены окружностями

$\triangleright_3$ Не составляя таблицы, изобразите на прямой все положительные ЧПТ из множества $\mathbb{F}_1(4, 2, -1, 1)$.

Приведённые данные наглядно демонстрируют следующие важные свойства, общие для всех ЧПТ.

1°. Отсутствие нуля: $0 \notin \mathbb{F}_1$.

2°. ЧПТ распределены на числовой прямой *неравномерно*.

3°. Чем больше модуль $\xi \in \mathbb{F}_1$, тем больше и расстояние между ξ и соседними элементами $\mathbb{F}_1$.

4°. Между нулём и минимальным положительным $\xi_{\min} \in \mathbb{F}_1$ существует «зазор», ширина которого больше расстояния от $\xi_{\min}$ до следующего ЧПТ в k раз.

$\triangleright_4$ Чему в общем случае равно k?

Двоичные ЧПТ выгодно отличаются от остальных тем, что в их нормализованной записи первый разряд d_0 всегда равен 1, поэтому его в памяти можно не хранить. *Таким образом для хранения p-разрядной двоичной мантиссы нам достаточно $(p-1)$ битов.*

1.3 Способы округления

Понятно, что если представление x в β-ичной системе счисления содержит больше p значащих цифр, мы не можем точно записать его в виде (1.1). В этом случае можно лишь приблизить x некоторым ЧПТ, которое в дальнейшем будем обозначать $\mathrm{R}(x)$.

Правилом округления для данного множества $\mathbb{F} \subset \mathbb{R}$ будем называть отображение

$$\mathrm{R} : \mathbb{R} \to \mathbb{F}$$

такое, что $\mathrm{R}(x) = x$, если $x \in \mathbb{F}$, и $\mathrm{R}(x) \approx x$ в противном случае.

Рассмотрим несколько способов задания R, считая $\beta = 10,\ p = 3$.

1. Отбрасывание «лишних» знаков ($\mathrm{R} = \mathrm{R}_d$): $\mathrm{R}_d(0.12345) = 1.23 \times 10^{-1}$.
2. Округление вверх («школьное округление», $\mathrm{R} = \mathrm{R}_u$). $\mathrm{R}_u(543.21) = 5.43 \times 10^2$, $\mathrm{R}_u(5678) = 5.68 \times 10^3$. В случае, когда запись x заканчивается на 5, его округляют до большего ЧПТ (вверх) : $\mathrm{R}_u(23.45) = 2.35 \times 10^1$.
3. Округление до чётного ($\mathrm{R} = \mathrm{R}_e$). Этот способ отличается от предыдущего только трактовкой «спорного» случая, когда x находится ровно между двумя ЧПТ $\underline{x}$ и $\overline{x}$. Оба эти приближения на самом деле равноправны, поэтому вместо того, чтобы всегда выбирать $\overline{x}$, мы будем с 50% вероятностью брать $\mathrm{R}_e(x) = \underline{x}$ либо $\mathrm{R}_e(x) = \overline{x}$.

   Реализовать это можно, всегда выбирая из $\underline{x}$ и $\overline{x}$ то, *мантисса которого заканчивается на чётную цифру*. Таким образом, получаем: $\mathrm{R}_e(23.45) = 2.34 \times 10^1$, но $\mathrm{R}_e(23.55) = 2.36 \times 10^1$.

Возникает вопрос: какой из описанных способов округления лучше? Ответ на него даёт следующая теорема.

ТЕОРЕМА 1.1. *Пусть x и y — два ЧПТ из $\mathbb{F}_1$. Рассмотрим последовательность $\{x_i\}$, определённую по правилу*

$$x_0 = x, \quad x_{i+1} = \mathrm{R}(\mathrm{R}(x_i + y) - y).$$

Если $\mathrm{R} = \mathrm{R}_e$, *то либо* $x_i = x\ \forall i \geqslant 0$, *либо* $x_i = x_1\ \forall i \geqslant 1$.

$\triangleright_5$ Пусть $\beta = 10,\ p = 3$. Приведите пример таких x и y, для которых утверждение теоремы не выполняется при $\mathrm{R} = \mathrm{R}_u$. Выпишите явную формулу для членов получившейся последовательности $\{x_i\}$.

$\triangleright_6$ Пусть $\beta = 2,\ p = 4$. Для чисел $x = 101.111,\ 11.101$ и 0.00010111 вычислите $\mathrm{R}_u(x)$ и $\mathrm{R}_e(x)$.

Заметим, что по стандарту IEEE 754 в современных ЭВМ используется $\mathrm{R} = \mathrm{R}_e$.

1.4 Расширение множества чисел с плавающей точкой

Глядя на рис. 1, мы видим, что для практического использования машинной арифметики нам не достаточно множества нормализованных ЧПТ $\mathbb{F}_1$. Как минимум к этому множеству нужно добавить нуль (см. свойство 1). Кроме этого, современные машинные арифметики включают также специальные значения для обозначения бесконечностей, результатов некорректных операций и т. п.

1.4.1 Денормализованные числа

Наличие «зазора» между нулём и минимальным положительным нормализованным ЧПТ (свойство 4) может привести к серьёзным проблемам на практике. Рассмотрим, например, ЧПТ $x = 0.75$ и $y = 0.625$ из рассмотренного модельного множества $\mathbb{F}_1(2, 3, -1, 2)$ (табл. 2). Так как $x - y = 0.125$, при любом разумном способе округления мы имеем $\mathrm{R}(x - y) = 0$. То есть, например, выполнение обычного кода типа

```
if (x != y) then z = 1/(x - y)
```

в нашем случае приведёт к плачевному результату.

Эта проблема в современных машинных арифметиках решается дополнением множества нормализованных ЧПТ так называемыми денормализованными числами.

Вещественные числа вида

$$0.d_1 d_2 \ldots d_{p-1} \times \beta^{e_{\min}},$$

где d_i — произвольные β-ичные цифры, называются *денормализованными* числами с плавающей точкой. Множество всех ДЧПТ с параметрами β, p, $e_{\min}$ будем обозначать $\mathbb{F}_0(\beta, p, e_{\min})$ либо кратко $\mathbb{F}_0$.

Введением денормализации мы сразу решаем две проблемы: получаем свойство

$$x = y \;\Leftrightarrow\; \mathrm{R}(x - y) = 0 \quad \text{для любых ЧПТ } x, y,$$

а также добавляем нуль ко множеству машинных чисел. Заметим, что для хранения денормализованных ЧПТ необходимо одно дополнительное значение для экспоненты (как правило это $e_{\min} - 1$).

▷7 Дорисуйте денормализованные числа на рис. 1.

1.4.2 Специальные величины

Стандарт IEEE 754, которому соответствуют практически все современные ЭВМ, предусматривает наличие специальных значений для машинных чисел, которым соответствуют не ЧПТ, а другие объекты. Простейшие объекты такого типа — это $+\infty$ и $-\infty$ (присутствуют также $+0$ и -0). Результаты вычислений с бесконечностями являются вполне определёнными: например, если x — положительное число, то по стандарту $x/\pm\infty = \pm 0$, $x/\pm 0 = \pm\infty$ и т. д. Кроме этого, стандартом определяются так называемые не-числа (NaN-ы, от «not a number»), которые обозначают результаты некорректных арифметических операций, таких как, например, извлечение корня из отрицательного числа.

1.4.3 Определение машинной арифметики

Машинными числами будем называть элементы множества

$$\mathrm{M} = \mathbb{F}_0 \cup \mathbb{F}_1 \cup \{+\infty, -\infty\}.$$

Машинной арифметикой с плавающей точкой (МАПТ) $\mathcal{A}$ будем называть множество машинных чисел M в совокупности с правилом округления R: $\mathcal{A} = \{\mathrm{M}, \mathrm{R}\}$.

При вычислениях в МАПТ будем считать, что результаты операций сложения, вычитания, умножения и деления являются *точно округляемыми*. Это означает, что результат указанных операций всегда вычисляется *точно*, после чего округляется до ЧПТ по правилу R.

1.5 Качественные характеристики машинной арифметики

Параметры $\beta, p, e_{\min}, e_{\max}$ и R полностью определяют свойства МАПТ, однако их знание не даёт прямой информации о том, насколько хороша или плоха соответствующая арифметика. С практической точки зрения пользователю нужны критерии, по которым можно определить качество МАПТ. Основным показателем качества будем считать точность, с которой арифметика приближает вещественные числа.

Абсолютной погрешностью округления для числа $x \in \mathbb{R}$ в данной МАПТ называется число

$$\Delta(x) = |x - \mathrm{R}(x)|, \tag{1.2}$$

а *относительной погрешностью округления* — число

$$\delta(x) = \frac{|x - \mathrm{R}(x)|}{|x|} = \frac{\Delta(x)}{|x|}. \tag{1.3}$$

Иногда относительную погрешность измеряют в процентах, умножая её на 100. Важно понимать, что при работе с машинной арифметикой уместнее всего оперировать относительными погрешностями, так как чем больше модуль x, тем больше $\Delta(x)$ (см. свойство 3).

Единицей в последней позиции (ulp — unit in the last place) для МАПТ называется функция $\mathrm{ulp} : \mathbb{R} \to \mathbb{R}$, определённая по следующему правилу. Если число $x \in \mathbb{R}$ лежит между двумя ЧПТ a и b и не равняется ни одному из них, то $\mathrm{ulp}(x) = |b - a|$. В противном случае $\mathrm{ulp}(x)$ равно расстоянию между двумя ближайшими к x ЧПТ.

ТЕОРЕМА 1.2 (Главное свойство $\mathrm{ulp}(x)$).

$$\Delta(x) \leqslant \mathrm{ulp}(x) \quad \textit{для} \quad \mathrm{R} = \mathrm{R}_d;$$
$$\Delta(x) \leqslant \mathrm{ulp}(x)/2 \quad \textit{для} \quad \mathrm{R} = \mathrm{R}_u, \mathrm{R}_e.$$

$\triangleright_8$ Докажите теорему.

$\triangleright_9$ Почему функция ulp так называется?

$\triangleright_{10}$ Постройте график $\mathrm{ulp}(x)$ для ЧПТ из таблицы 2.

Машинным эпсилон ε_M для МАПТ называется наименьшее положительное ε, удовлетворяющее условию

$$\mathrm{R}(1 + \varepsilon) > 1.$$

ТЕОРЕМА 1.3 (Главное свойство ε_M). *Для* $\mathrm{R} = \mathrm{R}_d, \mathrm{R}_u, \mathrm{R}_e$ *справедливо*

$$\delta(x) \leqslant \varepsilon_M \quad \textit{если} \quad \xi_{\min} \leqslant |x| \leqslant \xi_{\max},$$

где $\xi_{\min}$ *и* $\xi_{\max}$ *— минимальное и максимальное положительное нормализованное ЧПТ соответственно.*

$\triangleright_{11}$ Доказажите теорему.

$\triangleright_{12}$ Найдите ε_M для модельной арифметики из таблицы 2.

$\triangleright_{13}$ Выразите ε_M через $\beta,\ p,\ e_{\min},\ e_{\max}$ для различных R.

$\triangleright_{14}$ Будет ли справедлива теорема 1.3 если в качестве множества ЧПТ взять $\mathbb{F}_1 \cup \mathbb{F}_0$ а $\xi_{\min}$ — минимальное положительное в $\mathbb{F}_0$?

1.6 Стандарт IEEE 754

Международный стандарт «IEEE 754 floating point standard» определяет правила организации машинной арифметики с плавающей точкой. В настоящее время ему соответствует большинство вычислительных машин. В частности, наиболее распространённый тип данных, известный как double precision floating point (тип double в C/C++) по стандарту имеет следующие параметры:

β	p	$e_{\min}$	$e_{\max}$	N_{exp}	N_{total}	R
2	53	-1022	1023	11	64	R_e

$\triangleright_{15}$ Вычислите наименьшее положительное (де)нормализованное, наибольшее положительное (де)нормализованное и ε_M для данной арифметики.

$\triangleright_{16}$ Оцените $\Delta(0.1)$ и $\delta(0.1)$ в данной арифметике.

1.7 Трудности машинных вычислений

Потеря значимости. Эта проблема возникает при вычитании двух близких чисел, которые не являются точно представимыми в виде ЧПТ.

Приведём пример на модельной арифметике с $\mathbb{F}_1(2,3,-1,3)$ и $\mathrm{R} = \mathrm{R}_u$: пусть $x = 4.51,\ y = 4.49$. Имеем $\mathrm{R}(x) = 5,\ \mathrm{R}(y) = 4,$ и $\mathrm{R}(\mathrm{R}(x) - \mathrm{R}(y)) = 1$, тогда как $x - y = 0.02$. Таким образом мы имеем относительную погрешность вычисления равную 5000%, несмотря на то, что относительная погрешность округления для x и y составляет менее 12.5%. Отметим, что сложение этих двух чисел выполняется в данной арифметике точно.

Неассоциативность арифметических операций. При работе с машинными числами всегда следует помнить о том, что порядок операций существенно влияет на результат. Простейший случай — нарушение привычного свойства ассоциативности: если a, b и c — машинные числа, а $\circ$ — бинарная операция, то в общем случае $\mathrm{R}((a \circ b) \circ c) \neq \mathrm{R}(a \circ (b \circ c))$.

$\triangleright_{17}$ Приведите пример нарушения ассоциативности сложения в модельной арифметике $\mathbb{F}_1(2,3,-1,2)$ (рис. 1).

Резюме

Итак, при использовании машинной арифметики вычислитель всегда должен помнить о том, что как только он записывает в память ЭВМ число x, оно автоматически превращается в число $\tilde{x} = \mathrm{R}(x)$, которое *почти всегда* не будет равно x. Кроме того, чем больше модуль x, тем больше может быть разница между x и $\tilde{x}$ (абсолютная погрешность $\Delta(x)$). Относительная же погрешность округления, согласно теореме 1.3, почти всегда ограничена величиной ε_M.

2 Обусловленность задачи

2.1 Корректные задачи

Постоянное присутствие ошибок округления при работе с машиной арифметикой предъявляет особые требования к вычислительным алгоритмам и требует дополнительного анализа решаемой задачи. Так как практически все числа представляются в ЭВМ с погрешностью, необходимо знать насколько решение чувствительно к изменениям параметров задачи.

> Задача называется *корректно поставленной*, или просто *корректной*, если её решение (а) существует, (б) единственно и (в) непрерывно зависит от начальных данных. Если нарушено хотя бы одно из этих условий, задачу называют *некорректной*.

Решение некорректных задач на ЭВМ — весьма серьёзная проблема. Если задача некорректна, то наличие малейшей погрешности в начальных данных (которая практически неминуемо произойдёт как только вы запишете эти данные в память) может кардинальным образом исказить решение.

Существует ещё один класс задач, формально являющихся корректными, но решения которых, тем не менее, тоже весьма плохо ведут себя при наличии погрешностей в начальных данных — это так называемые плохо обусловленные задачи. В общих чертах, плохо обусловленной называется задача, которая при маленькой *относительной* погрешности в начальных данных даёт большую *относительную* погрешность в решении. Упор на относительную погрешность делается потому, что, как мы знаем из предыдущего раздела, при округлении вещественного числа x до машинного $\mathrm{R}(x)$ абсолютная погрешность $\Delta(x)$ зависит от величины $|x|$, в то время как относительная погрешность $\delta(x)$ постоянна для данной МАПТ.

2.2 Число обусловленности

В дальнейшем как параметры, так и решения рассматриваемых задач будут векторами пространства $\mathbb{R}^n$. Для исследования обусловленности задач нужно измерять «величины» этих векторов, для чего мы будем использовать нормы.

$\triangleright_1$ Что такое норма вектора в произвольном линейном пространстве?

В дальнейшем мы будем активно пользоваться двумя векторными нормами: максимум-нормой $\|\cdot\|_\infty$ (ввиду её простоты) и евклидовой нормой $\|\cdot\|_2$ (ввиду её «естественности»). Напомним, что

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

Обе эти нормы являются частными случаями p-нормы, определяемой формулой

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}.$$

Введём также обозначение

$$\delta(u,v) = \frac{\|u-v\|}{\|u\|},$$

которое можно назвать «относительной нормой разности» или «относительной погрешностью» векторов u и v.

Рассмотрим некоторую функцию

$$f : X \to Y,$$

положив для простоты $X \subset \mathbb{R}^m$, $Y \subset \mathbb{R}^n$ (хотя в общем случае X и Y могут быть подмножествами любого линейного векторного пространства). Возьмём произвольный вектор $x \in X$ и рассмотрим задачу вычисления $y = f(x)$ в предположении, что данная задача корректна. Пусть $\tilde{x} \in X$ — «возмущённый» вектор начальных параметров, и $\tilde{y} = f(\tilde{x})$ — соответствующее «возмущённое» решение. Наша задача оценить, во сколько раз $\delta(y,\tilde{y})$ может быть больше $\delta(x,\tilde{x})$.

Числом обусловленности задачи вычисления $f(x)$ назовём число

$$\text{æ}(x) = \sup_{\tilde{x}\in M} \frac{\delta(y,\tilde{y})}{\delta(x,\tilde{x})} = \sup_{\tilde{x}\in M} \frac{\|f(x)-f(\tilde{x})\|}{\|f(x)\|} \cdot \frac{\|x\|}{\|x-\tilde{x}\|},$$

где $M \subset X$ — некоторая проколотая окрестность точки x. Если $\text{æ}(x)$ велико, задачу называют *плохо обусловленной*.

Итак, по определению имеем

$$\delta(f(x), f(\tilde{x})) \leqslant \text{æ}(x)\,\delta(x,\tilde{x}), \quad \forall\, \tilde{x} \in M.$$

Это означает, что чем больше $\text{æ}(x)$, тем больше чувствительность решения задачи к относительной погрешности в начальных условиях.

Замечание 2.1. Естественно, понятие «большое число обусловленности» относительно. Судить о величине обусловленности можно лишь в контексте той машинной арифметики, которую вы используете, а ещё точнее — в контексте машинного эпсилон ε_M, так как он ограничивает относительную погрешность округления.

$\triangleright_2$ Каким должно быть число обусловленности, чтобы в данной МАПТ задача считалась плохо обусловленной?

Рассмотрим несколько примеров.

Пример 2.1. Вычисление скалярного произведения двух векторов: $(u,v) = \sum_{i=1}^{n} u_i v_i$. Будем считать, что вектор u фиксирован, а параметром задачи выступает вектор v, то есть $f(v) = (u,v)$. Тогда

$$\delta(f(v), f(\tilde{v})) = \frac{|(u, v-\tilde{v})|}{|(u,v)|} = \frac{\|u\|\|v-\tilde{v}\|\cos\alpha}{\|u\|\|v\|\cos\varphi} \leqslant \text{æ}(v)\,\delta(v,\tilde{v}), \quad \text{где } \text{æ}(v) = \frac{1}{\cos\varphi},$$

а φ — угол между u и v. Мы видим, что число обусловленности $\text{æ}(v) \to \infty$ при $\cos\varphi \to 0$, то есть вычисление скалярного произведения *почти ортогональных векторов* является плохо обусловленной задачей.

Например, пусть $u = (-1, 1.01)^T$, $v = (1,1)^T$, $\|\cdot\| = \|\cdot\|_\infty$. Тогда $f(v) = 0.01$. Заменим теперь v на $\tilde{v} = (0.99, 1.01)^T$, внося в начальные данные относительную погрешность $\delta(v,\tilde{v})$=0.01 (это означает, что мы изменили «размер» вектора на один процент).Имеем $f(\tilde{v}) = 0.0301$, но $\delta(f(v), f(\tilde{v})) = 2.01$ (результат изменился на 201 %).

Пример 2.2. Вычисление значения многочлена. Пусть $P(a_0,\ldots,a_n,x) = \sum_{i=0}^{n} a_i x^i$. Рассмотрим задачу вычисления данного многочлена, считая фиксированными x и все коэффициенты a_i кроме какого-то одного — a_k, который и будем считать параметром. Вычислим относительную погрешность решения:

$$\delta(f(a_k), f(\tilde{a}_k)) = \frac{|P(a_0,\ldots,a_n,x) - P(a_0,\ldots,\tilde{a}_k,\ldots,a_n,x)|}{|P(a_0,\ldots,a_n,x)|} = \frac{|(a_k-\tilde{a}_k)x^k|}{|a_k||\sum_{i=0}^{n}\frac{a_i}{a_k}x^i|} =$$
$$= \text{æ}(a_k)\,\delta(a_k,\tilde{a}_k),$$

$$\text{где} \quad æ(a_k) = \frac{|a_k\, x^k|}{|\sum_{i=0}^{n} a_i x^i|}.$$

Видим, что рассматриваемая задача может быть плохо обусловлена в двух случаях: когда x близко к одному их корней многочлена P, или когда $x \gg 1$ и k достаточно велико. Численный пример:

$$p(x) = (x-2)^{10} = x^{10} - 20x^9 + 180x^8 - 960x^7 + 3360x^6 - 8064x^5 + \\ + 13440x^4 - 15360x^3 + 11520x^2 - 5120x + 1024,$$

$x = 3,\ k = 9,\ f(-20) = 1$. Изменим коэффициент $a_9 = -20$ на 0.01 (что составляет 0.05%), $\tilde{a}_9 = -19.99$, и получим $f(\tilde{a}_9) = 197.83$, что на 19683% больше $f(a_9)$.

2.3 Обусловленность СЛАУ

2.3.1 Операторные матричные нормы

Мы приступаем к рассмотрению задачи решения системы линейных алгебраических уравнений (СЛАУ) вида

$$Ax = b, \tag{2.1}$$

где A — квадратная матрица размерности n, $x, b \in \mathbb{R}^n$. Прежде, чем приступить к алгоритмам численного решения этой задачи, исследуем её обусловленность.

Как и в случае векторных параметров, нам нужно будет как-то измерять «величину» матрицы A. Делать это мы будем с использованием операторных (подчинённых, индуцированных) норм.

При работе с матрицами (по крайней мере в контексте линейной алгебры) всегда важно помнить, что вы на самом деле имеете дело не с табличкой из $n \times m$ вещественных чисел, а с *линейным оператором*, то есть с отображением $A : \mathbb{R}^m \to \mathbb{R}^n$, которое обладает свойством линейности

$$A(\alpha x + \beta y) = \alpha Ax + \beta Ay, \quad \forall\, x, y \in \mathbb{R}^n,\ \alpha, \beta \in \mathbb{R}.$$

Важность такой точки зрения следует хотя бы из того, что так называемое «правило умножения матриц», которые многие считают аксиомой, есть ни что иное, как алгоритм вычисления композиции линейных операторов. В таком контексте вопрос об определении «величины» матрицы сводится к определению нормы соответствующего линейного оператора.

Нормой линейного оператора A называют число

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \sup_{\|x\|=1} \|Ax\|.$$

Замечание 2.2. Мы видим, что норма оператора полностью определяется векторной нормой. То есть каждая векторная норма порождает (индуцирует) соответствующую ей операторную матричную форму (в этом случае говорят также, что матричная норма подчинена векторной). В дальнейшем мы без оговорок будем предполагать, что используемая матричная форма подчинена векторной.

Таким образом, норма оператора равна максимальному «коэффициенту растяжения»: она показывает, во сколько раз под его действием может увеличиться норма вектора. Поэтому *по определению* для любой операторной нормы имеем важное свойство

$$\|Ax\| \leqslant \|A\| \, \|x\|. \tag{2.2}$$

$\triangleright_3$ Докажите свойство $\|AB\| \leqslant \|A\| \, \|B\|$ для любых линейных операторов A и B.

Напомним как вычисляются матричные нормы, индуцированные векторными нормами $\|\cdot\|_\infty$ и $\|\cdot\|_2$.

Теорема 2.1. *Векторной максимум-нормой* $\|\cdot\|_\infty$ *индуцируется матричная норма, вычисляемая по правилу*

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

Данную норму будем называть (строчной) максимум-нормой.

Теорема 2.2. *Евклидовой векторной нормой* $\|\cdot\|_2$ *индуцируется матричная норма, вычисляемая по правилу*

$$\|A\|_2 = \max_{1 \leqslant i \leqslant n} \sqrt{\lambda_i},$$

где $\{\lambda_i\}_{i=1}^n$ *— собственные значения матрицы* A^*A. *Эту норму называют спектральной матричной нормой.*

$\triangleright_4$ Докажите теоремы 2.1 и 2.2.

2.3.2 Число обусловленности матрицы

Рассмотрим СЛАУ (2.1). Решение этой системы, очевидно, сводится к вычислению

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

Исследуем обусловленность этой задачи, считая параметром вектор правой части b. Действуем по общей схеме, описанной в пункте 2.2:

$$f(b) = A^{-1}b.$$

$$\frac{\delta(f(b), f(\tilde{b}))}{\delta(b, \tilde{b})} = \frac{\|A^{-1}(b-\tilde{b})\|}{\|A^{-1}b\|} \cdot \frac{\|b\|}{\|b-\tilde{b}\|} \leqslant \frac{\|A^{-1}\|\,\|b\|}{\|A^{-1}b\|} \leqslant$$
$$\leqslant \left[\begin{array}{c} \|b\| = \|AA^{-1}b\| \leqslant \|A\|\,\|A^{-1}b\| \Rightarrow \\ \Rightarrow \|A^{-1}b\| \geqslant \|A\|^{-1}\|b\| \end{array} \right] \leqslant \|A^{-1}\|\,\|A\|. \qquad (2.3)$$

$\triangleright_5^{\bigstar}$ Для произвольной квадратной матрицы A и вектора b найдите возмущённый вектор $\tilde{b}$, при котором достигается наибольшее увеличение относительной погрешности в решении СЛАУ (2.1).

Исследование обусловленности относительно погрешностей в матрице A требует более тонкого подхода. Итак, пусть $f(A) = A^{-1}b$, вектор b считаем неизменным. Предположим, что возмущённая матрица $\tilde{A}$ может быть представлена в виде $\tilde{A} = A + \varepsilon H$, где H — некоторая матрица, определяющая «направление» изменения, $0 \leqslant \varepsilon$ — вещественный параметр, контролирующий величину этого изменения. Таким образом имеем

$$\delta(A, \tilde{A}) = \varepsilon \frac{\|H\|}{\|A\|}.$$

Так как $\tilde{A}$ является функцией ε, то решение $\tilde{x} = f(\tilde{A})$ тоже будет зависеть от параметра:

$$(A + \varepsilon H)\tilde{x}(\varepsilon) = b. \qquad (2.4)$$

Дифференцируя (2.4) по ε, с учётом $\tilde{x}(0) = x = A^{-1}b$ получаем

$$\tilde{x}'(0) = -A^{-1}Hx.$$

Теперь разложим $\tilde{x}(\varepsilon)$ по формуле Тейлора в точке $\varepsilon = 0$:

$$\tilde{x}(\varepsilon) = x + \varepsilon\tilde{x}'(0) + O(\varepsilon^2) = x - \varepsilon A^{-1}Hx + O(\varepsilon^2),$$

откуда

$$\delta(f(A), f(\tilde{A})) = f(x, \tilde{x}(\varepsilon)) \leqslant \varepsilon\|A^{-1}\|\,\|H\| + O(\varepsilon^2) =$$
$$= \|A\|\,\|A^{-1}\|\delta(A, \tilde{A}) + O(\varepsilon^2). \qquad (2.5)$$

Сопоставляя результаты (2.3), (2.5) получаем следующее определение.

Числом обусловленности невырожденной матрицы A называется число

$$æ(A) = \|A\|\,\|A^{-1}\|. \qquad (2.6)$$

Если матрица A вырождена, её число обусловленности полагается равным бесконечности.

Замечание 2.3. Несмотря на то, что мы ассоциируем это определение с матрицей A, необходимо чётко понимать, что речь идёт об обусловленности задачи решения СЛАУ.

$\triangleright_6$ Исследуйте обусловленность задачи умножения матрицы на вектор.

Замечание 2.4. Число обусловленности по определению зависит от нормы. В случаях, когда это необходимо, мы будем употреблять говорящие обозначения $æ_2(A)$ и $æ_\infty(A)$.

Свойства числа обусловленности матрицы

1°. $æ(A) \geqslant 1: \quad 1 = \|A^{-1}A\| \leqslant \|A\| \, \|A^{-1}\|$.

2°. $æ(AB) \leqslant æ(A)æ(B): \|AB\| \, \|(AB)^{-1}\| \leqslant \|A\| \, \|A^{-1}\| \, \|B\| \, \|B^{-1}\|$.

3°. Если $A = A^*$, то $æ_2(A) = \dfrac{|\lambda_{\max}|}{|\lambda_{\min}|}$, где $\lambda_{\min}$ и $\lambda_{\max}$ — минимальное и максимальное по модулю собственные значения матрицы A соответственно.

Замечание 2.5. Отметим, что в общем случае отсутствует прямая связь между величинами собственных значений и числом обусловленности. Например, собственные значения матрицы $A = \begin{pmatrix} 1 & \alpha \\ 0 & 1 \end{pmatrix}$ равны 1, в то время как $æ_2(A) \to \infty$ при $|\alpha| \to \infty$.

Замечание 2.6. Укажем на одно часто встречающееся заблуждение. Так как $æ(A)$ является своеобразным индикатором близости матрицы A к вырожденной, может возникнуть впечатление, что чем меньше определитель, тем больше число обусловленности. На самом же деле такой связи нет: достаточно заметить, что у A и A^{-1} взаимно обратные определители, но одинаковые числа обусловленности.

3 Метод Гаусса

3.1 Базовый метод Гаусса

Рассмотрим СЛАУ

$$Ax = b, \quad \det A \neq 0. \tag{3.1}$$

Один из способов решения этой системы заключается в переходе к эквивалентной системе (то есть, к системе с тем же решением) вида

$$Vx = g, \tag{3.2}$$

решение которой легко находится, если, например, V — верхне- или нижнетреугольная матрица. Решение системы (5.1), очевидно, может быть легко получено с помощью так называемой процедуры обратной подстановки (backsubstitution), или обратного хода. Например, для верхнетреугольной матрицы V эта процедура выглядит так:

$$x_i = \frac{1}{v_{ii}} \left(g_i - \sum_{j=i+1}^{n} v_{ij} x_j \right), \quad i = n, n-1, \ldots, 2, 1. \tag{3.3}$$

Переход от (3.1) к (5.1) осуществляется путём последовательного применения к обеим частям (3.1) некоторых линейных преобразований T_k:

$$T_N \ldots T_2 T_1 A x = T_N \ldots T_2 T_1 b,$$

то есть

$$V = \underbrace{T_N \ldots T_2 T_1}_{T} A, \quad g = Tb.$$

Если в качестве T_k использовать элементарные преобразования, то получим *метод Гаусса*.

$\triangleright_1$ Какие преобразования матриц называются элементарными?

Введём следующие обозначения: $\underline{a}_i$ — i-я строчка матрицы A, $[A]_k$ — матрица, составленная из k первых строк и k первых столбцов матрицы A.

Базовый алгоритм метода Гаусса

1: **for** $k = \overline{1, n-1}$ **do** `// Прямой ход метода`
2: **for** $i = \overline{k+1, n}$ **do**
3: $\underline{a}_i \leftarrow \underline{a}_i - \dfrac{a_{ik}}{a_{kk}} \underline{a}_k$
4: $b_i \leftarrow b_i - \dfrac{a_{ik}}{a_{kk}} b_k$
5: **end for**
6: **end for**
7: **for** $k = \overline{n, 1}$ **do** `// Обратный ход`
8: $x_k \leftarrow \dfrac{1}{a_{kk}} (b_k - \sum_{j=k+1}^{n} a_{kj} x_j)$
9: **end for**

$\triangleright_2$ Подсчитайте количество операций умножения в алгоритме.

Этап алгоритма, определяемый строками 2-5, будем называть *k-м шагом метода Гаусса*. На этом этапе с помощью элементарных преобразований обнуляются элементы k-го столбца, находящиеся ниже главной диагонали. В дальнейшем матрицу системы перед выполнением k-го

шага будем обозначать $A^{(k)}$ ($A^{(1)} = A$). Переход от матрицы $A^{(k)}$ к $A^{(k+1)}$ можно представить в виде $A^{(k+1)} = G_k A^{(k)}$, где

$$G_k = \left[\begin{array}{ccc|c|ccc} 1 & & & & & & \\ & \ddots & & & & & \\ & & & 1 & & & \\ & & & \alpha_{k+1}^{(k)} & 1 & & \\ & & & \alpha_{k+2}^{(k)} & & \ddots & \\ & & & \vdots & & & \ddots \\ & & & \alpha_n^{(k)} & & & 1 \end{array}\right]. \qquad (3.4)$$

(k-й столбец)

Матрица G_k представляет собой композицию $n-k$ элементарных преобразований второго типа.

$\triangleright_3$ Разложите G_k на элементарные множители.

$\triangleright_4$ Вычислите число обусловленности $æ_\infty(G_k)$.

Если M — произвольная квадратная матрица размерности n, то для вычисления $G_k M$ нужно для всех i от $k+1$ до n к i-й строке матрицы M прибавить k-ю, умноженную на $\alpha_i^{(k)}$. Согласно алгоритму метода Гаусса (см. строку 3) имеем

$$\alpha_i^{(k)} = -a_{ik}^{(k)} / a_{kk}^{(k)}, \qquad (3.5)$$

где $a_{ij}^{(k)}$ — элементы матрицы $A^{(k)}$. Очевидно, что если хотя бы один из элементов

$$\theta_k = a_{kk}^{(k)}$$

равен нулю, то прямой ход в базовом МГ неосуществим. В дальнейшем элементы θ_k будем называть *главными* или *ведущими*. Если же все главные элементы отличны от нуля, то приведённый алгоритм выполнится успешно.

Теорема 3.1. *Базовый алгоритм метода Гаусса осуществим тогда и только тогда, когда все главные угловые миноры матрицы A не равны нулю:* $|[A]_k| \neq 0 \quad \forall k = \overline{1,n}$.

Доказательство. Воспользуемся методом математической индукции. Преобразование G_1 корректно определено и первый шаг метода Гаусса выполним если (и только если) $a_{11}^{(1)} = a_{11} = |[A]_1| \neq 0$.

Пусть выполнимо k шагов. Это означает, что существует матрица $\tilde{G}_k$,

$$\tilde{G}_k = G_k G_{k-1} \ldots G_1, \quad \text{и} \quad A^{(k+1)} = \tilde{G}_k A.$$

$\triangleright_5$ Покажите, что матрицы $\tilde{G}_k$ имеют блочный вид

$$\left[\begin{array}{c|c} L_k & 0 \\ \hline \boxtimes & I \end{array}\right],$$

где L_k — нижнетреугольная матрица размерности k с единицами на главной диагонали, I — единичная матрица размерности $n-k$.

Запишем равенство $\tilde{G}_k A = A^{(k+1)}$ в блочном виде:

$$\underbrace{\left[\begin{array}{c|c} L_k & 0 \\ \hline \boxtimes & I \end{array}\right]}_{\tilde{G}_k} A = \underbrace{\left[\begin{array}{c|c} U_k & \boxtimes \\ \hline 0 & \boxtimes \end{array}\right]}_{A^{(k+1)}}, \tag{3.6}$$

где U_k — верхнетреугольная матрица размерности k. Критерием осуществимости $(k+1)$-го шага является условие

$$\theta_{k+1} = a^{(k+1)}_{k+1,k+1} \neq 0,$$

которое гарантирует существование G_{k+1} (см. (3.4), (3.5)). Из (3.6) имеем

$$[\tilde{G}_k]_{k+1}[A]_{k+1} = [A^{(k+1)}]_{k+1}.$$

Так как $|[\tilde{G}_k]_{k+1}| \neq 0$ и θ_{k+1} — единственный ненулевой элемент в последней строке матрицы $[A^{(k+1)}]_{k+1}$, то

$$\theta_{k+1} \neq 0 \;\Leftrightarrow\; |[A]_{k+1}| \neq 0.$$

■

3.2 Связь метода Гаусса и LU-разложения

LU-разложением невырожденной матрицы A называется её представление в виде

$$A = LU,$$

где L — нижнетреугольная матрица с единицами на главной диагонали, U — верхнетреугольная матрица.

ТЕОРЕМА 3.2 (Следствие теоремы 3.1). *Базовый алгоритм метода Гаусса для СЛАУ* (3.1) *выполним тогда и только тогда, когда существует LU-разложение матрицы A.*

Доказательство. $\boxed{\Rightarrow}$ Пусть осуществим базовый алгоритм. Тогда из формулы (3.6) при $k = n-1$ получаем $\tilde{G}_{n-1}A = A^{(n)}$, причём по построению $\tilde{G}_{n-1}$ — нижнетреугольная с единичной главной диагональю, а $A^{(n)}$ — верхнетреугольная. Отсюда получаем $A = LU$, где $L = (\tilde{G}_{n-1})^{-1}$, $U = A^{(n)}$.

$\triangleright_6$ Покажите, что $(\tilde{G}_{n-1})^{-1}$ — нижнетреугольная с единичной диагональю.

$\triangleright_7$ Докажите необходимость.

■

3.3 Метод Гаусса с выбором главного элемента

Для того, чтобы выполнение алгоритма метода Гаусса не обрывалось при $\theta_k = 0$ (и не только), *перед каждым шагом* метода применяется процедура, называемая *выбором главного элемента*. Суть процедуры: путём перестановки строк или столбцов матрицы $A^{(k)}$ поставить на позицию (k,k) ненулевой элемент. При этом, чтобы не «испортить» структуру матрицы, можно использовать лишь последние $n-k$ строк и $n-k$ столбцов. Существует несколько способов выбора главного элемента.

По столбцу: среди элементов $a_{ik}^{(k)}$ для i от k до n выбирается ведущий элемент $a_{i^*k}^{(k)}$, после чего переставляются местами строки k и i^*.

По строке: среди элементов $a_{kj}^{(k)}$ для j от k до n выбирается ведущий элемент $a_{kj^*}^{(k)}$, после чего переставляются местами *столбцы* k и j^*.

$\triangleright_8$ Как перестановка столбцов отразится на решении СЛАУ?

По матрице: среди элементов $a_{ij}^{(k)}$ для i,j от k до n выбирается ведущий элемент $a_{i^*j^*}^{(k)}$, после чего переставляются местами *строки* k и i^* и *столбцы* k и j^*.

Рассмотрим следующие вопросы.

1. Из каких соображений выбирать главный элемент?
2. Какой способ выбора главного элемента лучше?

Для ответа рассмотрим ещё раз матрицу G_k (3.4). Имеем

$$æ_\infty(G_k) = (1 + \max_i |\alpha_i^{(k)}|)^2, \tag{3.7}$$

откуда с учётом свойства 2 числа обусловленности имеем

$$æ_\infty(A^{(n)}) = æ_\infty(\tilde{G}_{n-1}A) \leqslant æ_\infty(A) \prod_{k=1}^{n-1} (1 + \max_i |\alpha_i^{(k)}|)^2.$$

Таким образом, даже если $æ(A)$ невелико, матрица $A^{(n)}$ может стать плохо обусловленной в случае больших значений $|\alpha_i^{(k)}|$. То есть, *сам процесс метода Гаусса может «испортить» исходную систему.*

Для исправления ситуации мы должны минимизировать величины (3.7). С учётом (3.5), получаем следующие ответы.

1. Главный элемент должен быть максимальным по модулю среди всех рассматриваемых.

2. Выбор главного элемента по столбцу оптимален по соотношению «качество/скорость».

$\triangleright_9$ Обоснуйте ответы.

$\triangleright_{10}$ Во сколько раз может возрасти $æ_\infty(A^{(n)})$ по сравнению с $æ_\infty(A)$ при оптимальном выборе главного элемента?

$\triangleright_{11}$ Покажите, что если метод Гаусса с выбором главного элемента не осуществим, то матрица системы вырождена.

3.4 Матричные уравнения

Метод Гаусса естественным образом обобщается на случай матричных уравнений вида

$$AX = B, \tag{3.8}$$

где A, как и ранее, — квадратная матрица порядка n, B — матрица размеров $n \times m$, X — неизвестная матрица тех же размеров, что и B. Возможно два подхода к решению таких уравнений.

1. Система (3.8) эквивалентна набору из m СЛАУ вида

$$Ax_j = b_j, \quad j = \overline{1, m},$$

где x_j и b_j — столбцы матриц X и B. Применять метод Гаусса к каждой такой системе нерационально (почему?), поэтому для их решения используется метод LU-разложения (см. следующий раздел).

2. Матричный метод Гаусса. Для того, чтобы адаптировать построенный выше алгоритм к решению матричных уравнений, достаточно строку 4 заменить на

$$\underline{b}_i \leftarrow \underline{b}_i - \frac{a_{ik}}{a_{kk}}\underline{b}_k,$$

а также модифицировать алгоритм обратной подстановки.

$\triangleright_{12}$ Проделайте это.

3.5 Обращение матрицы и вычисление определителя

Обращение матрицы эквивалентно решению матричного уравнения

$$AX = I,$$

где I — единичная матрица. Для решения этого уравнения могут использоваться оба описанных выше способа.

Определитель матрицы также вычисляется с помощью метода Гаусса:

$$\tilde{G}_{n-1}A = A^{(n)} \;\Rightarrow\; 1 \cdot |A| = |A^{(n)}| = a_{11}^{(n)} a_{22}^{(n)} \dots a_{nn}^{(n)}.$$

Однако, нужно помнить про важный нюанс: если в ходе метода переставлялись строки и столбцы, то каждая такая операция *меняла знак определителя на противоположный*. Поэтому окончательная формула такова:

$$|A| = (-1)^p a_{11}^{(n)} a_{22}^{(n)} \dots a_{nn}^{(n)}, \tag{3.9}$$

где p — количество перестановок строк и столбцов в ходе метода.

3.6 Метод прогонки

Рассмотрим СЛАУ

$$\begin{bmatrix} d_1 & e_1 & & & & \\ c_2 & d_2 & e_2 & & & \\ & c_3 & d_3 & e_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & c_{n-1} & d_{n-1} & e_{n-1} \\ & & & & c_n & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}. \tag{3.10}$$

Матрицы такой структуры называются *трёхдиагональными*. В приложениях достаточно часто встречаются такие системы. Особая структура этой матрицы позволяет найти решение системы методом Гаусса за $O(n)$ операций.

Алгоритм метода прогонки

for $k = \overline{2, n}$ **do** `// Прямой ход`
  $d_k \leftarrow d_k - e_{k-1}c_k/d_{k-1}$
  $b_k \leftarrow b_k - b_{k-1}c_k/d_{k-1}$
end for
$x_n = b_n/d_n$
for $k = \overline{n-1, 1}$ **do** `// Обратный ход`
  $x_k \leftarrow (b_k - e_k x_{k+1})/d_k$
end for

При выполнении алгоритма прогонки мы лишены возможности выбора главного элемента, так как при этом нарушилась бы трёхдиагональная структура матрицы A. Следовательно, метод прогонки осуществим тогда и только когда, когда все главные миноры матрицы отличны от нуля.

Существует также более простое для проверки достаточное условие осуществимости метода прогонки. Для его доказательства нам понадобятся следующие предварительные сведения.

Кругом Гершгорина D_i для квадратной матрицы A называется замкнутый круг на комплексной плоскости с центром в точке a_{ii} и радиусом

$$\rho_i = \sum_{j \neq i} |a_{ij}|.$$

ТЕОРЕМА 3.3 (Гершгорин). *Каждое собственное значение матрицы A лежит в одном из кругов Гершгорина.*

$\triangleright_{13}$ Докажите теорему.

Если элементы матрицы A удовлетворяют условиям

$$|a_{ii}| \geqslant \sum_{j \neq i} |a_{ij}| \quad \forall i = \overline{1, n}, \tag{3.11}$$

то говорят, что такая матрица обладает свойством *диагонального преобладания*. Если неравенство в (3.11) строгое, говорят о *строгом диагональном преобладании*.

ТЕОРЕМА 3.4. *Если матрица обладает свойством строгого диагонального преобладания, то все её главные миноры отличны от нуля.*

Доказательство. По условию имеем

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \quad \forall i = \overline{1, n}.$$

Отсюда $|[A]_1| = a_{11} \neq 0$. Далее рассмотрим $[A]_k$. Эта матрица, очевидно, тоже обладает строгим диагональным преобладанием. Следовательно, радиус круга Гершгорина D_i для $[A]_k$ меньше, чем $|a_{ii}|$, поэтому $0 \notin D_i \ \forall i = \overline{1, k}$. Значит, по теореме Гершгорина все собственные значения $[A]_k$ отличны от нуля, и $|[A]_k| \neq 0$. ■

ТЕОРЕМА 3.5 (Следствие теоремы 3.4). *Если матрица системы* (3.10) *удовлетворяет условиям*

$$|d_1| > |e_1|, \quad |d_i| > |c_i| + |e_i| \quad \forall i = \overline{2, n-1}, \quad \textit{и} \quad |d_n| > |c_n|,$$

то алгоритм прогонки выполним.

Резюме

- Метод Гаусса является наиболее экономичным в своём классе.
- Данный метод обычно используется для решения СЛАУ с полными матрицами (с малым процентом нулей) не очень большой размерности ($n \leqslant 1000$).
- В ходе метода Гаусса в общем случае ухудшается обусловленность решаемой системы.
- Для минимизации этого эффекта необходимо на каждом шаге выбирать главный элемент по столбцу (или по всей матрице).

4 LU-разложение

4.1 Базовый алгоритм LU-разложения

Рассмотрим последовательность СЛАУ

$$Ax = b^{(i)}, \quad i = \overline{1, N}. \tag{4.1}$$

и предположим, что векторы $b^{(i)}$ неизвестны заранее и поступают *по одному*, то есть мы не можем свести (4.1) к матричному уравнению. При решении каждой такой СЛАУ методом Гаусса будет тратиться $O(n^3)$ операций, причём к матрице A будут применяться *одни и те же* преобразования G_k.

Поэтому разумнее, однажды проделав прямой ход (или его аналог), построить LU-разложение $A = LU$, и в дальнейшем вычислять x путём решения двух СЛАУ с треугольными матрицами:

$$LUx = b \quad \Leftrightarrow \quad \begin{cases} Ly = b, \\ Ux = y. \end{cases} \tag{4.2}$$

$\triangleright_1$ Запишите алгоритм вычисления x по формулам (4.2).

$\triangleright_2$ Сравните общие вычислительные затраты при решении (4.1) методом Гаусса и методом LU-разложения.

Рассмотрим алгоритм построения LU-разложения в предположении,

что $|[A]_k| \neq 0 \ \ \forall k = \overline{1,n}$. По определению имеем

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \ell_{21} & 1 & 0 & \cdots & 0 \\ \ell_{31} & \ell_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \ell_{n3} & \cdots & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} u_{11} & u_{12} & u_{13} & \ldots & u_{1n} \\ 0 & u_{22} & u_{23} & \ldots & u_{2n} \\ 0 & 0 & u_{33} & \ldots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & u_{nn} \end{bmatrix}}_{U} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} & \ldots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \ldots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \ldots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \ldots & a_{nn} \end{bmatrix}}_{A}. \tag{4.3}$$

При машинной реализации алгоритма матрицы L и U будем хранить на месте матрицы A:

$$A \leftarrow \tilde{A} = \underbrace{\begin{bmatrix} u_{11} & u_{12} & u_{13} & \ldots & u_{1n} \\ \ell_{21} & u_{22} & u_{23} & \ldots & u_{2n} \\ \ell_{31} & \ell_{32} & u_{33} & \ldots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \ell_{n3} & \cdots & u_{nn} \end{bmatrix}}_{L-I+U}, \quad \text{т. е.} \quad \tilde{a}_{ij} = \begin{cases} u_{ij}, & i \leqslant j, \\ \ell_{ij}, & i > j. \end{cases}$$

Из (4.3) имеем

$$a_{ij} = \sum_{k=1}^{n} \ell_{ik} u_{kj} = \sum_{k=1}^{\min(i,j)} \ell_{ik} u_{kj}. \tag{4.4}$$

Выделяя последние слагаемые в суммах (4.4), получаем

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} \ell_{ik} u_{kj} \qquad \text{при } i \leqslant j; \tag{4.5}$$

$$\ell_{ij} = \frac{1}{u_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} \ell_{ik} u_{kj} \right) \qquad \text{при } i > j, \tag{4.6}$$

или

$$\tilde{a}_{ij} = \begin{cases} a_{ij} - \displaystyle\sum_{k=1}^{i-1} \tilde{a}_{ik} \tilde{a}_{kj} & \text{при } i \leqslant j; \\ \dfrac{1}{\tilde{a}_{jj}} \left(a_{ij} - \displaystyle\sum_{k=1}^{j-1} \tilde{a}_{ik} \tilde{a}_{kj} \right) & \text{при } i > j. \end{cases} \tag{4.7}$$

Таким образом, неизвестные элементы матриц L и U последовательно выражаются через a_{ij} и уже найденные ℓ_{ik} и u_{kj}.

1: **for** $j = \overline{1,n}$ **do**
2: **for** $i = \overline{1,n}$ **do**
3: $a_{ij} \leftarrow a_{ij} - \sum_{k=1}^{\min(i,j)-1} a_{ik}a_{kj}$
4: **if** $i > j$ **then**
5: $a_{ij} \leftarrow a_{ij}/a_{jj}$
6: **end if**
7: **end for**
8: **end for**

$\triangleright_3$ Оцените число мультипликативных операций в алгоритме.

$\triangleright_4$ Будет ли полученная в итоге матрица U совпадать с матрицей $A^{(n)}$ из базового метода Гаусса? Докажите.

4.2 Выбор главного элемента

По аналогии с методом Гаусса, этап алгоритма LU-разложения, определяемый циклом в строках 2-7 будем называть j-м шагом LU-разложения. Для того, чтобы алгоритм был универсальным, необходимо реализовать выбор главного элемента $\tilde{a}_{jj} = u_{jj}$, на который происходит деление в строке 5.

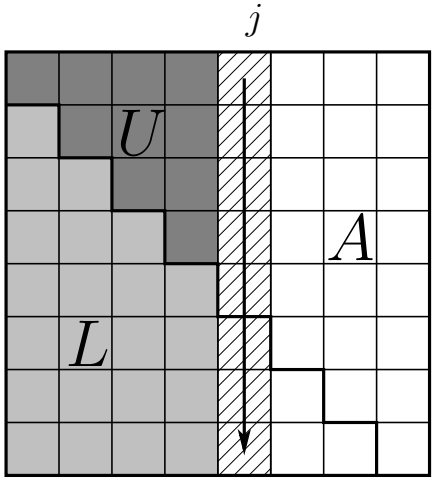


Рисунок 2. Вид матрицы системы перед j-м шагом алгоритма LU-разложения.

Рассмотрим матрицу $A^{(j)}$, которая получается из A после $(j-1)$-го шага разложения (рис. 2). К этому моменту столбцы с 1-го по $(j-1)$-й уже содержат часть матриц L и U, а оставшиеся столбцы являются столбцами исходной матрицы A. Имеем ли мы право переставлять в этой «составной» матрице строки и если да, то какие? Строки с 1 по $(j-1)$-ю

переставлять нельзя, иначе нарушится структура матрицы $\tilde{A}$. Перестановка же строк с j по n эквивалентна преобразованию

$$LU = A \quad \mapsto \quad PLU = PA,$$

где P — матрица перестановки.

$\triangleright_5$ Какой вид имеет матрица P?

Итак, на j-м шаге алгоритма мы имеем право переставлять строки с номерами от j до n. Поэтому элементы u_{ij} для i от 1 до $j-1$, вычисляемые по формуле (4.5), можно найти сразу. После этого нужно осуществить перестановку строк, но проблема в том, что ведущий элемент u_{jj} *ещё неизвестен*, как неизвестны и возможные кандидаты на его место.

Поэтому перестановка должна быть выполнена таким образом, *чтобы элемент* $a_{jj}^{(j)} = u_{jj}$, вычисляемый по формуле

$$u_{jj} = a_{jj} - \sum_{k=1}^{j-1} \ell_{jk} u_{kj} \tag{4.8}$$

был максимальным по модулю. Заметим, что элементы ℓ_{ij} вычисляются по формуле (4.6), которая при $i = j$ отличается от (4.8) только множителем $1/u_{jj}$. Поэтому выбор главного элемента на j-ом шаге LU-разложения осуществляется следующим образом.

1. Вычисляем кандидатов на роль ведущего элемента: для всех i от j до n вычисляем $a_{ij}^{(j)} = \tilde{a}_{ij}$ по второй формуле из (4.7), только *без деления на* $\tilde{a}_{jj}$;
2. Среди полученных значений $a_{ij}^{(j)}$ для $i \geqslant j$ выбираем максимальный по модулю $a_{i^*j}^{(j)}$;
3. Меняем местами j-ю и i^*-ю строки матрицы $A^{(j)}$;
4. Для всех i от $j+1$ до n делим $a_{ij}^{(j)}$ на $a_{jj}^{(j)}$.

Замечание 4.1. Для того, чтобы после получения LU-разложения корректно решить СЛАУ (4.2), необходимо предварительно переставить элементы вектора b в соответствии с перестановками, которые происходили в ходе разложения. Поэтому стандартная процедура должна возвращать не только матрицу $\tilde{A}$, но и вектор перестановок p. Кроме этого, для корректного вычисления определителя необходимо возвращать $s = \pm 1$ — значение чётности числа перестановок.

$\triangleright_6$ Можно ли по одному только вектору p определить s?

4.3 Разложение Холецкого

ТЕОРЕМА 4.1 (Разложение Холецкого). *Пусть A — самосопряжённая матрица над полем $\mathbb{C}$: $A = A^*$. Если все главные миноры $|[A_k]|$ отличны от нуля, то существует разложение*

$$A = R^* D R, \tag{4.9}$$

где R — верхнетреугольная матрица, $D = \operatorname{diag}(d_1, d_2, \ldots, d_n)$, $|d_k| = 1$ $\forall k = \overline{1,n}$. Формула (4.9) называется разложением Холецкого (Cholesky).

Доказательство. Так как $|[A]_k| \neq 0$, по теореме 3.2 существует LU-разложение

$$A = LU = U^* L^* = A^*,$$

откуда $L = U^*(L^* U^{-1}) = U^* H$, и

$$A = LU = U^* H U. \tag{4.10}$$

Рассмотрим матрицу $H = L^* U^{-1}$. С одной стороны H — верхнетреугольная, так как является произведением верхнетреугольных матриц L^* и U^{-1}. С другой стороны $H = (U^*)^{-1} L$, то есть H является ещё и нижнетреугольной. Следовательно,

$$H = \operatorname{diag}(h_1, h_2, \ldots, h_n).$$

Положим $d_k = h_k/|h_k|$, $\tilde{H} = \operatorname{diag}(\sqrt{|h_1|}, \sqrt{|h_1|}, \ldots, \sqrt{|h_n|})$. Тогда (4.10) даёт

$$A = U^* H U = U^*(\tilde{H}^* D \tilde{H}) U = (\tilde{H} U)^* D (\tilde{H} U) = R^* D R,$$

что и требовалось доказать.

$\triangleright_7$ Докажите, что матрица D существует, т. е. все $h_k \neq 0$.

■

Квадратная матрица A над полем $\mathbb{R}$ ($\mathbb{C}$) называется положительно определённой ($A > 0$), если

$$(Ax, x) > 0 \quad \forall x \in \mathbb{R}^n\ (\mathbb{C}^n),\ x \neq 0.$$

В комплексном случае мы подразумеваем, что все (Ax, x) вещественны.

Свойства положительно определённых матриц

1°. Если $A > 0$ то $|[A]_k| \neq 0$ $\forall k = \overline{1,n}$.

2°. Если $A^* = A$, то $A > 0 \Leftrightarrow$ все собственные значения A вещественны и положительны.

ТЕОРЕМА 4.2 (следствие теоремы 4.1). *Если $A = A^*$ и $A > 0$, то существует разложение*

$$A = R^* R,$$

где R — верхнетреугольная матрица.

Доказательство. Согласно свойству 1 для матрицы A существует разложение (4.9). Значит, нам достаточно показать, что матрица D — единичная. По условию имеем меем

$$(Ax, x) = (R^* DRx, x) = (DRx, Rx) > 0 \quad \forall x \neq 0.$$

Так как R невырождена, $\forall y \in \mathbb{C}^n \quad \exists\, x : \; y = Rx$, то есть

$$(Dy, y) = \sum_{i=1}^{n} d_i y_i \bar{y}_i > 0 \quad \forall y \neq 0. \tag{4.11}$$

Возьмём в качестве y k-й единичный орт: $y_i = \delta_{ik}$. Тогда с учётом того, что $|d_k| = 1$, из (4.11) получаем $d_k = 1 \;\; \forall k = \overline{1, n}$. ■

Таким образом, в случае вещественной матрицы A теоремы 4.1 и 4.2 влекут следующие утверждения: если A симметричная ($A = A^T$) и все $|[A]_k| \neq 0$, то существует разложение вида

$$A = R^T DR, \tag{4.12}$$

где R — вещественная верхнетреугольная, D — диагональная матрица с элементами ± 1 на диагонали. Если к тому же $A > 0$, то $D = I$.

4.4 Метод квадратного корня

Методом квадратного корня называется метод решения вещественной СЛАУ $Ax = b$ с симметричной матрицей A путём построения разложения Холецкого

$$A = R^T DR.$$

Обозначим $L = R^T$, $U = DR$. Тогда аналогично методу LU-разложения имеем

$$a_{ij} = \sum_{k=1}^{\min(i,j)} \ell_{ik} u_{kj} = \left[\begin{array}{c} \ell_{ik} = r_{ki}, \\ u_{kj} = d_k r_{kj} \end{array} \right] = \sum_{k=1}^{\min(i,j)} d_k r_{ki} r_{kj}. \tag{4.13}$$

В силу симметрии достаточно рассмотреть (4.13) только для верхнего треугольника матрицы A ($i \leqslant j$):

$$i = j: \quad d_i r_{ii}^2 = a_{ii} - \sum_{k=1}^{i-1} d_k r_{ki}^2 = \omega_i \;\Rightarrow\; \left[\begin{array}{l} d_i = \operatorname{sign} \omega_i, \\ r_{ii} = \sqrt{|\omega_i|}; \end{array} \right. \tag{4.14a}$$

$$i < j: \quad r_{ij} = \frac{1}{d_i r_{ii}} \left(a_{ij} - \sum_{k=1}^{i-1} d_k r_{ki} r_{kj} \right). \tag{4.14b}$$

Детали программной реализации

1. Так как матрица A симметрична, достаточно хранить в памяти только её верхний треугольник.
2. Аналогично случаю LU-разложения расчётные формулы (4.14) позволяют последовательно (построчно) находить элементы матрицы R и хранить их на месте исходной матрицы: $A \leftarrow R$.
3. Решение получаемой в итоге СЛАУ $R^T D R x = b$ осуществляется путём применения двух обратных подстановок: $R^T y = b$, затем $Rx = Dy$ (так как $D = D^{-1}$).

$\triangleright_8$ Запишите алгоритм прямого и обратного хода метода квадратного корня.

$\triangleright_9$ Оцените число мультипликативных операций в прямом ходе метода и сравните с методом Гаусса.

Резюме

- Метод LU-разложения фактически представляет собой «законсервированный» метод Гаусса.
- Особенно удобен этот метод при решении большого количества СЛАУ с одной и той же матрицей A.
- При реализации LU-разложения кроме матриц L и U необходимо возвращать ещё и вектор перестановок, чтобы впоследствии корректно решить СЛАУ.
- Метод квадратного корня — частный случай LU-разложения для симметричных матриц.

5 Методы ортогональных преобразований

5.1 Метод отражений

Недостатком метода Гаусса и его модификаций является то, что элементарные преобразования G_k в общем случае ухудшают обусловленность исходной системы ($æ_\infty(G_k) > 1$). Поэтому разработаны альтернативные методы приведения матрицы A к треугольному виду, основанные на ортогональных преобразованиях.

$\triangleright_1$ Докажите, что если матрица Q ортогональна, то $æ_2(Q) = 1$.

Система из двух уравнений

Рассмотрим СЛАУ

$$Ax = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}. \tag{5.1}$$

Для её решения применим к обеим частям ортогональное линейное преобразование T, которое «обнуляет» элемент a_{21}:

$$TAx = A'x = \begin{bmatrix} a'_{11} & a'_{12} \\ 0 & a'_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b'_1 \\ b'_2 \end{bmatrix} = Tb. \tag{5.1′}$$

Пусть a_j — j-й столбец матрицы A, тогда $a'_j = Ta_j$. Так как ортогональные преобразования сохраняют евклидову норму векторов, имеем $\|a_j\| = \|a'_j\|$ (в дальнейшем $\|\cdot\| = \|\cdot\|_2$), поэтому $a'_{11} = \pm\|a_1\|$.

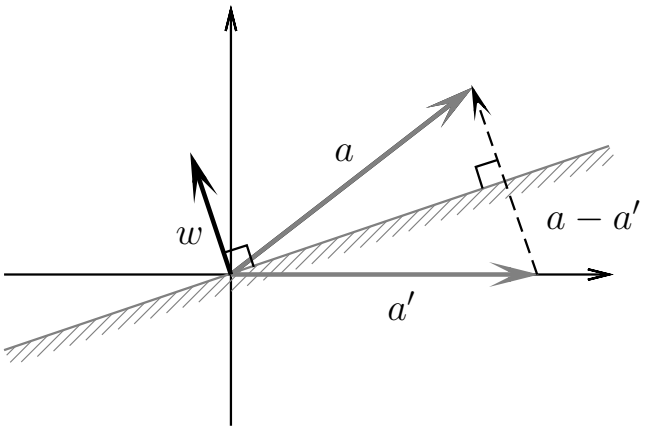


Рисунок 3. Преобразование отражения на плоскости.

Рассмотрим вектор $a = a_1$, под действием T переходящий в $a' = a'_1$. Наша задача — определить T как преобразование отражения относительно какой-то гиперплоскости, что равносильно нахождению вектора

нормали w к этой гиперплоскости, $\|w\| = 1$. Нетрудно заметить, что векторы $a - a'$ и w коллинеарны, следовательно

$$w = \pm\frac{a - a'}{\|a - a'\|}. \tag{5.2}$$

Теперь рассмотрим обратную задачу: по данным векторам a и w найти a'. Из (5.2) имеем $\pm\|a - a'\|w = a - a' \Rightarrow a' = a - \gamma w$, $\gamma \in \mathbb{R}$. Так как $(a + a') \perp w$, $(a + a', w) = (2a - \gamma w, w) = 0 \Rightarrow \gamma = 2(a, w)$, откуда в итоге получаем

$$a' = a - 2(a, w)w. \tag{5.3}$$

$\triangleright_2$ Докажите, что преобразование отражения является а) линейным и б) ортогональным.

Таким образом, в случае системы (5.1) прямой ход метода отражений состоит в следующем:

1. По формуле (5.2) найти вектор нормали w, определяющий гиперплоскость, при отражении относительно которой вектор a_1 переходит в $a'_1 = (\pm\|a_1\|, 0)^T$.
2. По формуле (5.3) применить найденное преобразование отражения к векторам a_2 и b.

Общая схема метода

В общем случае формулы (5.2), (5.3) тоже остаются в силе.

1. Вычислить w по формуле (5.2), где $a = a_1$, $a' = a'_1 = (\pm\|a_1\|, 0, \ldots, 0)^T$.
2. $a_1 \leftarrow a'_1$; вычислить $a_j \leftarrow a'_j \quad \forall\, j \neq 1$, $b \leftarrow b'$ по формуле (5.3).
3. Повторить шаги 1-2 для нижней правой подматрицы A и соответствующего подвектора b размерности $n - 1$, и так далее до тех пор, пока матрица A не станет верхнетреугольной.

$\triangleright_3$ При каких условиях указанный алгоритм выполним?

$\triangleright_4$ Как в ходе алгоритма распознать вырожденную матрицу A?

$\triangleright_5$ Сравните сложность метода отражений и метода Гаусса.

Замечание 5.1. При реализации метода отражений существует свобода выбора знака для вектора a'_1. Чтобы избежать вычитания близких чисел при вычислении w по формуле (5.2), этот знак выбирают таким образом, чтобы он был *противоположен знаку* a_{11} (a_{kk} в общем случае). Тогда при вычислении $a_1 - a'_1$ фактически будут *складываться* два одинаковых по модулю числа.

5.2 QR-разложение

Найдём в явном виде матрицу преобразования отражения, задаваемого формулой (5.3).

$$a' = a - 2(a,w)w = Ia - 2w(w^T a) = Ia - 2(ww^T)a = (I - 2ww^T)a.$$

Пусть $w \in \mathbb{R}^n(\mathbb{C}^n)$, $\|w\|_2 = 1$. Матрица

$$H = H(w) = I - 2ww^T$$

называется *матрицей отражения*. Она задаёт преобразование отражения относительно гиперплоскости с нормалью w.

Свойства матрицы отражения

1°. $H(w) = H(w)^{-1}$.

2°. Матрица отражения является симметричной (самосопряжённой).

3°. Матрица отражения является ортогональной (унитарной).

4°. Все собственные значения матрицы отражения равны ± 1.

$\triangleright_6$ Докажите свойства 1-3.

Докажем свойство 4. Так как матрица H унитарна, имеем $\|H\| = 1$. Пусть λ — произвольное собственное значение H, x — соответствующий собственный вектор: $Hx = \lambda x$. Тогда $\|Hx\| = \|x\| = \|\lambda x\|$, откуда $|\lambda| = 1$. Докажем теперь, что собственные числа самосопряжённой матрицы вещественны:

$$H = H^* \Rightarrow (Hx,x) = (x,H^*x) = (x,Hx) = \overline{(Hx,x)} \Rightarrow (Hx,x) \in \mathbb{R} \quad \forall\, x \in \mathbb{C}^n.$$

Далее пусть x — собственный вектор. Тогда

$$(Hx,x) = \lambda(x,x) = \lambda\|x\|^2 \;\Rightarrow\; \lambda = \frac{(Hx,x)}{\|x\|^2} \in \mathbb{R}.$$

Таким образом, $|\lambda| = 1$ и $\lambda \in \mathbb{R}$, то есть $\lambda = \pm 1$.

Замечание 5.2. Заметим, что вычисление преобразования отражения по формуле (5.3) требует $O(n)$ операций умножения и сложения, в то время как умножение на матрицу $H(w)$ требует $O(n^2)$ операций. Поэтому в явном виде $H(w)$ практически никогда не используют, а хранят только w.

Процесс преобразования A к верхнетреугольному виду по методу отражений можно представить в виде

$$A^{(k+1)} = Q_k A^{(k)},$$

где Q_k — блочная матрица вида

$$Q_k = \left[\begin{array}{c|c} I_{k-1} & 0 \\ \hline 0 & H(w_k) \end{array}\right], \tag{5.4}$$

I_{k-1} — единичная матрица размерности $k-1$, w_k — вектор нормали размерности $n-k+1$.

$\triangleright_7$ Докажите, что матрицы Q_k являются ортогональными.

$\triangleright_8$ Докажите, что если Q и H ортогональны, то Q^{-1} и QH тоже ортогональны.

Таким образом мы имеем

$$Q_{n-1} \dots Q_2 Q_1 A = \tilde{Q} A = A^{(n)},$$

откуда

$$A = QR, \qquad \text{где} \quad Q = \tilde{Q}^{-1},\ R = A^{(n)}. \tag{5.5}$$

Теорема 5.1 (о QR-разложении). *Для любой вещественной квадратной матрицы A существует QR-разложение: $A = QR$, где Q — ортогональная, R — верхнетреугольная матрица с неотрицательными диагональными элементами. Если* $\det A \neq 0$, *то все диагональные элементы R положительны.*

Доказательство. Доказательство напрямую следует из приведённых выше рассуждений. Если A вырождена, то в ходе метода отражений будем встречать нулевые векторы a_1. В этом случае нужно просто перейти к следующему шагу. ■

$\triangleright_9$ Обобщите теорему на случай прямоугольной матрицы A.

Рассмотрим подробно алгоритм построения QR-разложения методом отражений. Из (5.5) имеем

$$Q = \tilde{Q}^{-1} = (Q_{n-1} \dots Q_1)^{-1} = Q_1 \dots Q_{n-1},$$

где Q_k определяются формулой (5.4) (здесь мы использовали тот факт, что $Q_k^{-1} = Q_k$). Согласно замечанию 5.2 вместо того, чтобы хранить отдельно матрицу Q, мы можем хранить лишь векторы w_k, которые однозначно определяют Q. Кроме того, так как $w_k \in \mathbb{R}^{n-k+1}$, можно их хранить на месте нижнего треугольника матрицы A (для этого необходимо завести отдельный вектор для хранения диагональных элементов a_{kk}).

Предположим, что нам известно QR-разложение матрицы A. Тогда решение СЛАУ $Ax = b$ осуществляется за $O(n^2)$ операций:

$$QRx = b \quad \Leftrightarrow \quad \begin{cases} Qy = b, \\ Rx = y. \end{cases}$$

Таким образом, сначала вычисляется $y = Q^{-1}b = Q^T b$, а затем обратной подстановкой находится x.

Если матрица Q хранится в виде набора нормалей w_k, то вычисление вектора

$$y = Q^{-1}b = Q_{n-1} \dots Q_1 b$$

эквивалентно последовательному применению к вектору b всех преобразований отражения, использованных при построении QR-разложения.

$\triangleright_{10}$ Сравните такой подход с простым умножением на матрицу Q^T.

5.3 Метод вращений

Система из двух уравнений

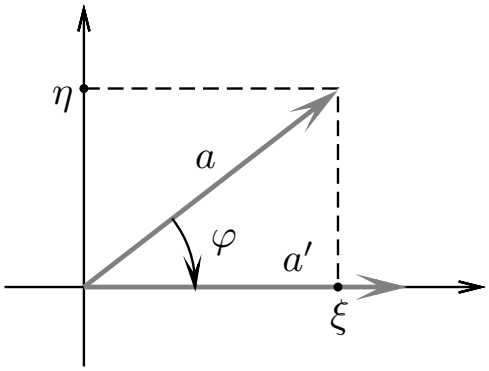


Рисунок 4. Преобразование вращения на плоскости.

Рассмотрим снова систему (5.1) и вектор $a = a_1$. Переход к системе (5.1′) будем теперь осуществлять с помощью преобразования вращения. Угол φ необходимо выбрать так, чтобы при вращении у вектора a обнулилась вторая координата. Координаты вектора a обозначим ξ и η. С помощью перехода к полярным координатам нетрудно показать, что матрица вращения на угол α *против часовой стрелки* имеет вид

$$V = V(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}. \quad (5.6)$$

$\triangleright_{11}$ Проделайте это.

В нашем случае матрица преобразования $T : A \mapsto A'$ равна $V(-\varphi)$, где φ — угол между вектором a и осью x_1:

$$T = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}, \quad (5.7)$$

$$c = \frac{\xi}{\sqrt{\xi^2 + \eta^2}}, \quad s = \frac{\eta}{\sqrt{\xi^2 + \eta^2}}. \quad (5.8)$$

Итак, метод вращений для системы (5.1) состоит в следующем:

1. Найти $c = \cos\varphi$ и $s = \sin\varphi$ по формуле (5.8), положив $\xi = a_{11}$, $\eta = a_{21}$.

2. $a_1 \leftarrow a_1' = (\|a_1\|, 0)^T$.
3. $a_2 \leftarrow a_2' = Ta,\ b \leftarrow b' = Tb$, где T — матрица (5.7).

Общий случай. В общем случае СЛАУ из n уравнений основное отличие метода вращений от метода отражений заключается в том, что для выполнения k-го шага метода нам необходимо выполнить $n-k$ операций вращения (а не одно преобразование, как в методе отражений), по одной на каждую обнуляемую координату. Рассмотрим матрицу

$$V_{ki} = \begin{array}{c} \begin{matrix} & & k & & i & \end{matrix} \\ \left[\begin{array}{c|c|c|c|c} \begin{matrix} 1 & & \\ & \ddots & \\ & & 1 \end{matrix} & & & & \\ \hline & c_{ki} & & s_{ki} & \\ \hline & & \ddots & & \\ \hline & -s_{ki} & & c_{ki} & \\ \hline & & & & \begin{matrix} 1 & & \\ & \ddots & \\ & & 1 \end{matrix} \end{array}\right] \end{array} \begin{matrix} \\ \\ k \\ \\ i \\ \\ \end{matrix} \qquad \begin{matrix} c_{ki} = \cos\varphi_{ki}, \\ s_{ki} = \sin\varphi_{ki}. \end{matrix} \tag{5.9}$$

Это матрица элементарного вращения в координатной плоскости (x_k, x_i) на угол φ_{ki} *по часовой стрелке*. Рассмотрим k-й шаг метода вращений (обнуление всех элементов a_{ik} для $i = \overline{k+1, n}$). Он состоит из $n-k$ частей, каждая из которых соответствует умножению на матрицы вида (5.9):

$$Q_k = V_{kn} V_{k,n-1} \dots V_{k,k+1}. \tag{5.10}$$

Например, для системы из трёх уравнений мы будем иметь

$$\overbrace{\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{23} & s_{23} \\ 0 & -s_{23} & c_{23} \end{bmatrix}}_{V_{23}} \underbrace{\begin{bmatrix} c_{13} & 0 & s_{13} \\ 0 & 1 & 0 \\ -s_{13} & 0 & c_{13} \end{bmatrix}}_{V_{13}}}^{Q_2} \overbrace{\underbrace{\begin{bmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{V_{12}}}^{Q_1} \underbrace{\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \end{bmatrix}}_{A^{(3)}}.$$

Элементы c_{ki} и s_{ki} вычисляются по формулам (5.8), где $\xi = a_{kk},\ \eta = a_{ik}$ — текущие (а не исходные, конечно) элементы матрицы A. Заметим, что умножение на матрицу V_{ki} изменяет в произвольном $x \in \mathbb{R}^n$ только

k-й и i-й элементы:

$$V_{ki}\begin{bmatrix} x_1 \\ \vdots \\ x_k \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ c\,x_k + s\,x_i \\ \vdots \\ -s\,x_k + c\,x_i \\ \vdots \\ x_n \end{bmatrix} \qquad (c = c_{ki}, s = s_{ki}).$$

Таким образом, i-я ($(i-k)$-я по счёту) часть k-го шага метода вращений описывается следующим алгоритмом.

Основная часть алгоритма метода вращений

1: $d \leftarrow \sqrt{a_{kk}^2 + a_{ik}^2}$; $\quad c \leftarrow a_{kk}/d$; $\quad s \leftarrow a_{ik}/d$;
2: $a_{kk} \leftarrow d$; $\quad a_{ik} \leftarrow 0$;
3: **for** $j = \overline{k+1, n}$ **do**
4: $\quad tmp \leftarrow c\,a_{kj} + s\,a_{ij}$;
5: $\quad a_{ij} \leftarrow -s\,a_{kj} + c\,a_{ij}$;
6: $\quad a_{kj} \leftarrow tmp$;
7: **end for**

▷12 Запишите алгоритм полностью и сравните сложность с методом Гаусса.

Замечание 5.3. *Матрицы V_{ki} в явном виде никогда не используются и в памяти не хранятся!*

С помощью метода вращений, очевидно, можно строить QR-разложение матрицы A: k-й шаг метода соответствует умножению на ортогональную матрицу (5.10), поэтому аналогично методу отражений имеем

$$A = (Q_{n-1} \dots Q_1)^{-1} A^{(n)} = QR.$$

▷13 Обдумайте способ оптимального хранения матрицы Q и оптимального решения СЛАУ $Qy = b$.

Резюме

- Методы отражений и вращений более устойчивы к ошибкам округления, но более трудоёмки, чем метод Гаусса.
- При реализации этих методов нет необходимости в перестановке строк или столбцов.
- QR-разложение матрицы A существует даже если матрица системы вырождена.

Задачи и упражнения

1. Для указанных значений $\{\beta, p, e_{\min}, e_{\max}\}$ изобразите на числовой прямой соответствующие множества нормализованных и денормализованных чисел с плавающей точкой, вычислите ε_M:

а) $\{2, 3, -2, 1\}$, б) $\{3, 2, -1, 1\}$, в) $\{5, 2, -1, 1\}$.

2. Известны три подряд идущих нормализованных машинных числа из некоторой двоичной арифметики с плавающей точкой: 3.75, 4, 4.5. Чему равен ε_M, если округление в арифметике осуществляется до ближайшего машинного числа?

3. Пусть ε_M — машинный эпсилон в некоторой двоичной машинной арифметике с плавающей точкой. Оцените абсолютную погрешность округления $\Delta(x)$ для

а) $x = 0.1$, б) $x = \pi$, в) $x = 123.4$, г) $x = 123456.7$.

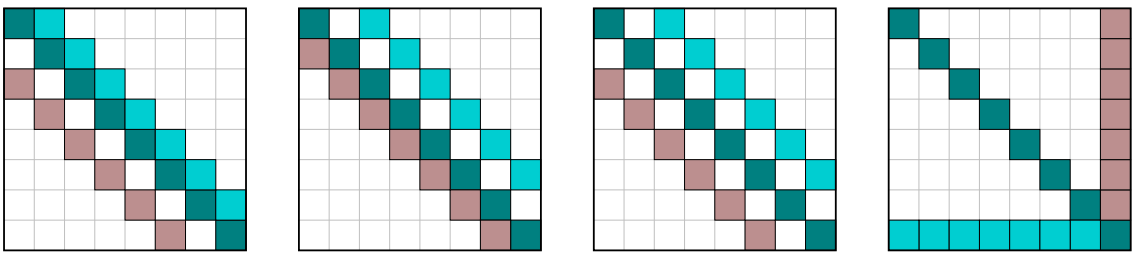
4. Пусть ε_M — машинный эпсилон в некоторой β-ичной машинной арифметике с плавающей точкой. Оцените абсолютную погрешность округления $\Delta(x)$ для произвольного x.

5. Рассмотрим плохо обусловленную СЛАУ $Ax = b$. Всегда ли наличие погрешности в векторе b означает большую погрешность в решении? Приведите пример.

6. Исследуйте обусловленность задачи вычисления определителя матрицы размерности 2 относительно погрешности, вносимой в элемент на позиции $(1, 1)$. Приведите примеры хорошо и плохо обусловленной задачи такого типа.

7. Исследуйте обусловленность задачи вычисления определителя произвольной квадратной матрицы A относительно погрешности, вносимой в элемент a_{ij}. Приведите примеры хорошо и плохо обусловленной задачи такого типа.

8. По аналогии с методом прогонки постройте алгоритм решения СЛАУ с матрицей указанной структуры. Матрица задаётся в виде трёх векторов c, d и e. Варианты:



9. Методом LU-разложения решите СЛАУ

а) $\left[\begin{array}{ccc|c} 4 & -3 & 0 & 1 \\ 4 & -4 & -2 & 0 \\ 16 & -14 & -1 & 2 \end{array}\right]$, б) $\left[\begin{array}{ccc|c} 3 & -1 & 4 & 1 \\ -3 & -2 & -9 & -6 \\ -6 & 5 & -4 & 2 \end{array}\right]$, в) $\left[\begin{array}{ccc|c} 1 & 1 & -1 & 1 \\ -4 & -3 & 3 & -4 \\ -3 & 1 & -3 & -5 \end{array}\right]$.

10. Постройте LU-разложение матрицы вида

$$\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}.$$

11. Дана трёхдиагональная матрица A, как обычно задаваемая тремя векторами c, d и e. Запишите алгоритм построения LU-разложения для такой матрицы, а также алгоритм решения СЛАУ $Ax = b$ с помощью построенного разложения. Оцените сложность алгоритмов.

12. Решите методом квадратного корня СЛАУ

а) $\left[\begin{array}{ccc|c} 9 & 3 & 3 & 0 \\ 3 & 5 & -5 & 10 \\ 3 & -5 & 19 & -24 \end{array}\right]$, б) $\left[\begin{array}{ccc|c} 4 & -2 & 8 & 2 \\ -2 & 5 & 2 & 3 \\ 8 & 2 & 34 & 10 \end{array}\right]$,

в) $\left[\begin{array}{cccc|c} -9 & -3 & 3 & 3 & 0 \\ -3 & -2 & 3 & 1 & -1 \\ 3 & 3 & 11 & -13 & -10 \\ 3 & 1 & -13 & 24 & 25 \end{array}\right]$, г) $\left[\begin{array}{cccc|c} -4 & 6 & -2 & 8 & 2 \\ 6 & -18 & 6 & -21 & -3 \\ -2 & 6 & 7 & 7 & 1 \\ 8 & -21 & 7 & -9 & 12 \end{array}\right]$.

13. Постройте вычислительный алгоритм решения СЛАУ $Ax = b$ с вещественной симметричной матрицей A, основанный на модифицированном разложении Холецкого: $A = LDL^T$, где L — нижнетреугольная матрица с единицами на главной диагонали, D — диагональная матрица с ненулевыми элементами.

14. Дана симметричная трёхдиагональная матрица A, задаваемая вектором d (главная диагональ) и вектором c (диагональ над и под главной). Запишите алгоритм построения разложения Холецкого $A = R^T DR$ для такой матрицы, а также алгоритм решения СЛАУ $Ax = b$ с помощью построенного разложения. Оцените сложность алгоритмов.

15. Методом отражений решите СЛАУ

а) $\left[\begin{array}{ccc|c} \frac{2}{3} & 0 & 1 & \frac{5}{3} \\ \frac{2}{3} & -\frac{14}{5} & \frac{13}{5} & \frac{7}{15} \\ \frac{1}{3} & -\frac{2}{5} & \frac{9}{5} & \frac{26}{15} \end{array}\right]$, б) $\left[\begin{array}{ccc|c} -2 & -\frac{34}{15} & -\frac{14}{3} & -\frac{8}{3} \\ 1 & -\frac{13}{15} & -\frac{8}{3} & -\frac{11}{3} \\ 2 & -\frac{10}{3} & -\frac{1}{3} & -\frac{7}{3} \end{array}\right]$.

16. Методом вращений решите СЛАУ

а) $\left[\begin{array}{ccc|c} \frac{1}{\sqrt{2}} & \frac{1}{2} & 1+\frac{1}{\sqrt{2}} & -1 \\ \frac{1}{\sqrt{2}} & \frac{1}{2}+\sqrt{2} & 1+\frac{1}{\sqrt{2}} & -1 \\ -1 & -1+\frac{1}{\sqrt{2}} & -1+\sqrt{2} & -\sqrt{2} \end{array}\right]$, б) $\left[\begin{array}{ccc|c} 0 & \sqrt{2} & 2\sqrt{2} & \sqrt{2} \\ -1 & -\frac{4}{3} & \frac{11}{3} & 5 \\ 2\sqrt{2} & -\frac{\sqrt{2}}{3} & -\frac{4\sqrt{2}}{3} & -\sqrt{2} \end{array}\right]$.

17. Рассмотрим вектор $w = \frac{1}{\sqrt{n}}(1, \ldots, 1)^T \in \mathbb{R}^n$ и матрицу $A = I - 2ww^T$. Найдите A^{2010}.

18. Рассмотрим матрицу $A = \begin{bmatrix} \frac{\sqrt{3}}{2} & 0 & -\frac{1}{2} \\ 0 & 1 & 0 \\ \frac{1}{2} & 0 & \frac{\sqrt{3}}{2} \end{bmatrix}$. Найдите A^{2010}.

19. Дана трёхдиагональная матрица A, как обычно задаваемая тремя векторами c, d и e. Запишите алгоритм построения QR-разложения такой матрицы а) методом отражений и б) методом вращений, а также алгоритм решения СЛАУ $Ax = b$ с помощью построенного разложения. Оцените сложность алгоритмов.

20. Постройте семейство матриц A_n, для которых методы решения СЛАУ, основанные на ортогональных преобразованиях, будут на практике работать существенно лучше, чем метод Гаусса.

Список литературы

[1] D. Goldberg, “What every computer scientist should know about floating-point arithmetic” // ACM Computing Surveys vol. 23 №1 (1991), pp. 5-48.

[2] Голуб Дж., Ван Лоун Ч. Матричные вычисления: Пер. с англ. — М.: Мир, 1999. — 548 с, ил.

[3] Д. К. Фаддеев, В. К. Фаддеева. Вычислительные методы линейной алгебры. — М.: Наука, 1963.

[4] Крылов В. И., Бобков В. В., Монастырный П. И. Начала теории вычислительных методов: Линейная алгебра и нелинейные уравнения. — Мн.: Наука и техника, 1985.

[5] Numerical Recipes in C: the art of scientific computing / William H. Press [et al.]. — 2nd ed. Cambridge University Press, 1992.

Учебное издание

Фалейчик Борис Викторович

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ АЛГЕБРЫ: БАЗОВЫЕ ПОНЯТИЯ И АЛГОРИТМЫ

Учебно-методическое пособие

В авторской редакции

Ответственный за выпуск *Б. В. Фалейчик*

Подписано в печать 04.05.2010. Формат 60 × 84/16. Бумага офсетная.
Гарнитура Roman. Усл. печ. л. 2,56. Уч.-изд. л. 2,28. Тираж 50 экз. Зак.

Белорусский государственный университет.
ЛИ №02330/0494425 от 08.04.2009.
Пр. Независимости, 4, 220030, Минск.

Отпечатано с оригинала-макета заказчика на копировально-множительной технике
факультета прикладной математики и информатики
Белорусского государственного университета.
Пр. Независимости, 4, 220030, Минск.