МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО
ОБРАЗОВАНИЯ
"САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА
(национальный исследовательский университет)"

А. И. ЖДАНОВ

ВВЕДЕНИЕ В ВЫЧИСЛИТЕЛЬНУЮ ЛИНЕЙНУЮ АЛГЕБРУ

Электронное учебное пособие

САМАРА

2011

Автор: **Жданов Александр Иванович**

Редакторская обработка И. В. Афутина
Компьютерная верстка И. В. Афутина
Доверстка Н. Ю. Лысенкова

Жданов, А. И. Введение в вычислительную линейную алгебру [Электронный ресурс] : электрон. учеб. пособие / А. И. Жданов; М-во образования и науки РФ, Самар. гос. аэрокосм. ун-т им. С. П. Королева (нац. исслед. ун-т). - Электрон. дан. (714,1 Кбайт). - Самара, 2011. -1 эл. опт. диск (CD-ROM).

Рассматриваются причины существенного различия между «теоретическим» и «компьютерным» алгоритмами. Изучается важнейшее понятие современной вычислительной математики – понятие устойчивости компьютерного алгоритма. Излагаются устойчивые компьютерные алгоритмы решения наиболее практически значимых задач вычислительной линейной алгебры.

Учебное пособие предназначено для подготовки бакалавров направления 010400.62 "Прикладная математика и информатика" факультета информатики, специализирующихся на задачах математического моделирования физических и информационных процессов, изучающих дисциплину "Вычислительная линейная алгебра" в 7 семестре.

Разработано на кафедре прикладной математики.

Оглавление

Предисловие

В учебном пособии изучаются существенные различия между "теоретическим" и "компьютерным" алгоритмами, выясняются причины такого различия, которые кроются в особенностях компьютерной арифметики с плавающей точкой.

В данном учебном пособии достаточно обстоятельно изучается арифметика чисел с плавающей точкой и особенности ее реализации на большинстве современных компьютеров. В последнее время эти аспекты компьютерной реализации вычислительных алгоритмов приобрели особую значимость в связи с развитием высокопроизводительных вычислительных систем.

В настоящее время наиболее универсальный подход представления действительных чисел в компьютере – IEEE-арифметика, основанная на представлении действительных чисел с плавающей точкой. В системе с плавающей точкой позиции десятичной или бинарной точки хранятся отдельно от самих цифр и промежутки между соседними представленными числами соответствуют пропорции с величинами цифр. В этом принципиальное отличие арифметики чисел с плавающей точкой от представления с фиксированной точкой, где все промежутки одинаковы.

Данное учебное пособие – подробное содержательное, но отнюдь не многословное пособие, по которому, с одной стороны, можно ознакомиться с принципами, положенными в основу устойчивых компьютерных алгоритмов вычислительной линейной алгебры, а с другой – научиться эти алгоритмы реализовывать, избегая при этом существенно искажающих результат погрешностей. Многие из этих погрешностей на первый взгляд столь незначительны, что, казалось бы, не должны оказывать никакого практического влияния на рассчитываемые величины.

В пособии подробно рассмотрены такие фундаментальные понятия компьютерных алгоритмов как, *устойчивость* и *обратная устойчивость*. Эти понятия дают возможность исследовать точность реальных

компьютерных алгоритмов.

Приведены достаточно эффективные новые вычислительные алгоритмы для решения плохо обусловленных и неполного ранга линейных задач наименьших квадратов, позволяющие их реализацию в виде конкретных устойчивых компьютерных алгоритмов.

Практическая ценность результатов, представленных в настоящем учебном пособии, также объясняется тем бесспорным фактом, что важнейшим алгоритмом вычислительной математики, используемым в качестве блока решения большинства практических вычислительных задач, является алгоритм решения СЛАУ, при построении решений которых принципиальным фактором является наличие детерминированных или случайных погрешностей задания исходных данных.

Учебное пособие содержит достаточное число упражнений, необходимых для более полного усвоения рассмотренного теоретического материала.

Глава 1

Вспомогательные сведения из линейной алгебры

1.1. Арифметические пространства

В вычислительных методах линейной алгебры под *линейным (векторным) пространством над полем вещественных чисел* понимается *арифметическое n-мерное линейное пространство* $\mathbb{R}^n$, элементами которого являются вектор-столбцы вида $x = (x_1, \ldots, x_n)^\top$, составленные из фиксированного числа n вещественных компонент (координат) $x_i \in \mathbb{R}$, $1 \le i \le n$, где $\top$ – знак транспонирования. Аналогично под *линейным (векторным) пространством над полем комплексных чисел* понимается *арифметическое n-мерное линейное пространство* $\mathbb{C}^n$, элементами которого являются вектор-столбцы вида $x = (x_1, \ldots, x_n)^\top$, составленные из фиксированного числа n комплексных компонент (координат) $x_i \in \mathbb{C}$, $1 \le i \le n$.

Если $x = (x_1, \ldots, x_n)^\top$ и $y = (y_1, \ldots, y_n)^\top$ – элементы $\mathbb{R}^n$ (или $\mathbb{C}^n$), а $\alpha \in \mathbb{R}$ (или $\alpha \in \mathbb{C}$), то сумма $x + y = (x_1 + y_1, \ldots, x_n + y_n)^\top$ и произведение на скаляр $\alpha x = (\alpha x_1, \ldots, \alpha x_n)^\top$ принадлежит пространству $\mathbb{R}^n(\mathbb{C}^n)$. Условия замкнутости относительно сложения векторов и умножения на скаляр характеризуют абстрактное понятие линейного пространства. При этом в линейном пространстве выполняется система аксиом:

$$x + y = y + x, \quad x + (y + z) = (x + y) + z,$$

$$x + 0 = x, \quad x + (-x) = 0,$$

$$\alpha(x + y) = \alpha x + \alpha y, \quad (\alpha + \beta)x = \alpha x + \beta x, \quad 1 \cdot x = x,$$

где $x, y, z \in \mathbb{R}^n\,(\mathbb{C}^n)$, $0 = (0, \ldots, 0)^\top$ – нулевой вектор пространства $\mathbb{R}^n\,(\mathbb{C}^n)$, $\alpha, \beta \in \mathbb{R}\,(\mathbb{C})$ и 1 – вещественные скаляры, $-x = (-1)x$. Как правило, вместо $x + (-y)$ пишут $x - y$. Символ 0, как обычно, будет обозначать нулевой скаляр, нулевой вектор или нулевую матрицу, т.е. структуры, состоящие только из нулевых элементов; тип структуры и ее размеры, если не указаны явно, определяются контекстом.

Аналогичным образом можно рассмотреть пространство $\mathbb{R}^n$ (или $\mathbb{C}^n$), образованное вектор-строками. В дальнейшем, если не оговорено иное, предполагается, что линейное пространство $\mathbb{R}^n$ (или $\mathbb{C}^n$) образовано вектор-столбцами.

Введем теперь важное понятие подпространства линейного пространства $\mathbb{R}^n$. В дальнейшем в этом разделе будет рассматриваться лишь пространство $\mathbb{R}^n$, хотя все результаты распространяются и на $\mathbb{C}^n$. Множество $\mathcal{L}$ векторов из $\mathbb{R}^n$ называется *подпространством* пространства $\mathbb{R}^n$, если оно замкнуто относительно сложения и умножения вектора на число (соответственно из $\mathbb{R}$ или $\mathbb{C}$), т.е. если

1° вместе с каждым x множество $\mathcal{L}$ содержит и все векторы вида αx $(\alpha x \in \mathcal{L})$,

2° вместе с векторами x, y множество $\mathcal{L}$ содержит их сумму $x + y$ $(x + y \in \mathcal{L})$.

Подпространство $\mathcal{L}$ может состоять лишь из одного вектора - нулевого. Такое подпространство назовем *тривиальным*. Нетривиальное подпространство, не совпадающее с $\mathbb{R}^n$, называется *собственным*. Например, все векторы из $\mathbb{R}^3$ вида $(x_1, x_2, x_3) = (\xi + 2\eta, 2\xi + \eta, 3\xi + 3\eta)$ образуют собственное подпространство $\mathbb{R}^3$.

Введем понятие базиса подпространства $\mathcal{L}$ пространства $\mathbb{R}^n$.

Линейной комбинацией векторов $x^{(1)}, x^{(2)}, \ldots, x^{(k)} \in \mathcal{L} \subset \mathbb{R}^n$ *с коэффициентами* α_k назовем вектор вида

$$x = \alpha_1 x^{(1)} + \alpha_2 x^{(2)} + \ldots + \alpha_k x^{(k)} = \sum_{j=1}^{k} \alpha_j x^{(j)}.$$

Система векторов $\{x^{(1)}, x^{(2)}, \ldots, x^{(k)}\}$ называется *линейно зависимой*, если можно подобрать такие коэффициенты $\alpha_1, \alpha_2, \ldots, \alpha_k$, среди которых хотя бы один отличен от нуля, что линейная комбинация $x = \sum_{j=1}^{k} \alpha_j x^{(j)}$ является нулевым вектором. Бесконечная система векторов из $\mathbb{R}^n$ называется *линейно зависимой*, если линейно зависима одна из ее конечных подсистем.

Система векторов $\{x^{(1)}, x^{(2)}, \ldots, x^{(k)}\}$ *линейно независима*, если равенство $\sum_{j=1}^{k} \alpha_j x^{(j)} = 0$ выполняется лишь при $\alpha_1 = \alpha_2 = \ldots = \alpha_k = 0$.

П р и м е р. Векторы из $\mathbb{R}^3$

$$x^{(1)} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \quad x^{(2)} = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}, \quad x^{(3)} = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$$

линейно зависимы, так как $x^{(1)} + x^{(2)} - 3x^{(3)} = 0$. В то же время $x^{(1)}$, $x^{(2)}$ линейно независимы, так как из векторного равенства $\alpha_1 x^{(1)} + \alpha_2 x^{(2)} = 0$ следует, что $\alpha_1 = \alpha_2 = 0$.

Легко убедиться, что система ненулевых попарно ортогональных векторов пространства $\mathbb{R}^n$ всегда линейно независима. Действительно, пусть $\sum_{j=1}^{m} \alpha_k x^{(k)} = 0$. Умножим это равенство скалярно на $x^{(j)}$, $1 \le j \le m$. Так как $(x^{(i)}, x^{(j)}) = 0$ при $i \ne j$, то получаем в результате уравнение $\alpha_j (x^{(j)}, x^{(j)}) = 0$, $1 \le j \le m$. Следовательно, $\alpha_j = 0$, $1 \le j \le m$.

Максимальная система линейно независимых векторов подпространства $\mathcal{L}$ называется базисом $\mathcal{L}$; другими словами, система линейно независимых векторов $\{x^{(1)}, x^{(2)}, \ldots, x^{(k)}\} \subset \mathcal{L}$ является базисом подпространства $\mathcal{L}$, если любой вектор $x \in \mathcal{L}$ представим в виде $x = \alpha_1 x^{(1)} + {} + \alpha_2 x^{(2)} + \ldots + \alpha_k x^{(k)}$.

Можно показать, что каждое нетривиальное подпространство $\mathcal{L}$ пространства $\mathbb{R}^n$ имеет базис и все базисы одного подпространства $\mathcal{L}$ состоят из одинакового числа векторов. Это число называется *размерностью* подпространства $\mathcal{L}$ и обозначается $\dim \mathcal{L}$. Размерность тривиального (нулевого) подпространства $\{0\}$ полагаем равной нулю.

Важно отметить, что каждое нетривиальное подпространство $\mathcal{L}$ имеет ортонормированный базис и что любая ортонормированная система векторов подпространства $\mathcal{L}$ может быть дополнена до ортонормированного базиса этого подпространства.

П р и м е р. В качестве $\mathcal{L}$ рассмотрим $\mathbb{R}^n$. Стандартный (ортонормированный) базис $\mathbb{R}^n$ - это набор n векторов $e^{(i)} = (0, \ldots, 0, 1, 0, \ldots, 0)^\top$, содержащих 1 в i-позиции и 0 в остальных. Поэтому $\dim \mathbb{R}^n = n$.

В силу определения базиса подпространства $\mathcal{L}$ пространства $\mathbb{R}^n$ любой вектор $x \in \mathcal{L}$ представим в виде $x = \sum_{j=1}^{k} \alpha_j x^{(j)}$, где $x^{(1)}, x^{(2)}, \ldots, x^{(k)}$ - базис подпространства $\mathcal{L}$. Коэффициенты α_j определены однозначно. Действительно, пусть $x = \sum_{j=1}^{k} \beta_j x^{(j)}$ - другое представление вектора x через базис $x^{(1)}, x^{(2)}, \ldots, x^{(k)}$. Тогда, очевидно, что

$$(\alpha_1 - \beta_1) x^{(1)} + (\alpha_2 - \beta_2) x^{(2)} + \ldots + (\alpha_k - \beta_k) x^{(k)} = 0.$$

Из линейной независимости векторов базиса $x^{(k)}$, $1 \leq j \leq k$, следует, что $\alpha_j - \beta_j = 0$ для всех $j = 1, 2, \ldots, k$. Коэффициенты α_j в разложении вектора $x = \sum_{j=1}^{k} \alpha_j x^{(j)}$ по базису $x^{(1)}, x^{(2)}, \ldots, x^{(k)}$ подпространства $\mathcal{L}$ называются *координатами вектора x в базисе* $x^{(1)}, x^{(2)}, \ldots, x^{(k)}$.

С помощью координат векторов в базисе $x^{(1)}, x^{(2)}, \ldots, x^{(k)}$ подпространство $\mathcal{L} \in \mathbb{R}^n$ можно отождествлять с пространством $\mathbb{R}^k$, т.е. любой вектор $x \in \mathcal{L} \in \mathbb{R}^n$ можно представить в виде $x = (x_1, \ldots, x_k)^\top$, где x_i - координаты x в базисе $x^{(1)}, x^{(2)}, \ldots, x^{(k)}$. Фактически и арифметическое n-мерное пространство $\mathbb{R}^n$ является множеством векторов с компонентами (координатами) разложения по стандартному базису $e^{(1)}, e^{(2)}, \ldots, e^{(n)}$.

Базис подпространства $\mathcal{L}$ пространства $\mathbb{R}^n$ дает возможность определить это подпространство конструктивно: множество векторов из $\mathcal{L}$ состоит из всех линейных комбинаций векторов базиса, другими словами, является *линейной оболочкой базиса*. Линейная оболочка любой системы векторов $x^{(1)}, x^{(2)}, \ldots, x^{(k)}$, обозначаемая через $\operatorname{span}(x^{(1)}, x^{(2)}, \ldots, x^{(k)})$, образует подпространство размерности m, $m \leq k$. Преимущество базиса перед другими системами векторов, которые имеют одинаковые линейные оболочки, заключается в том, что в разложении векторов по базису коэффициенты определяются однозначно.

Некоторые подпространства возникают естественным образом в связи с матрицами. Так, с $m \times n$-матрицей A связано *пространство столбцов* $\operatorname{im} A$ (или *образ матрицы*), т.е. линейная оболочка столбцов матрицы A,

$$\operatorname{im} A = \operatorname{span}(a_1, a_2, \ldots, a_n) = \{y \in \mathbb{R}^m : Ax = y, \forall\, x \in \mathbb{R}^n\},$$

где $A = (a_1, a_2, \ldots, a_n)$. *Пространство строк* матрицы A это $\operatorname{im} A^\top$.

Пространства строк и столбцов матрицы A имеют одинаковую размерность, называемую *рангом матрицы* и обозначаемую через $\operatorname{rank} A$. Говорят, что *матрица A размера $m \times n$ имеет неполный ранг*, если $\operatorname{rank} A < \min(m, n)$, и *полный ранг*, если $\operatorname{rank} A = \min(m, n)$. Заметим, что $\operatorname{rank} PAQ = \operatorname{rank} A$ для любых невырожденных матриц P и Q.

Можно показать, что вырожденность $n \times n$-матрицы A эквивалентна существованию такого ненулевого вектора x, что $Ax = 0$. Следовательно, квадратная матрица A порядка n невырождена, если $\operatorname{rank} A = n$, и вырождена, если $\operatorname{rank} A < n$.

Определим *правое нуль-пространство* $m \times n$-матрицы A (или *ядро матрицы*) как множество

$$\ker A = \{x \in \mathbb{R}^n : Ax = 0\}$$

и соответственно левое нуль-пространство матрицы A как $\ker A^\top = \{y \in \mathbb{R}^m : A^\top y = 0\}$. Несложно показать, что

$$\operatorname{rank} A = n - \dim \ker A = m - \dim \ker A^\top.$$

1.2. Матричная алгебра

1.2.1. Определения и обозначения. Приведем определения и обозначения из матричной алгебры используемые в данном учебном пособии.

$\mathbb{R}^{m\times n}$ ($\mathbb{C}^{m\times n}$) - множество вещественных (комплексных) $m \times n$-матриц, имеющих m строк и n столбцов;

a_{ij} ($i = 1, \dots, m$, $j = 1, \dots, n$) - элементы матрицы $A \in \mathbb{R}^{m\times n}$, $A = (a_{ij})$;

$0 \in \mathbb{R}^{m\times n}$ - матрица, состоящая из нулей (или $0_{m\times n}$);

$\operatorname{diag}(d_{11}, \dots, d_{nn}) \in \mathbb{R}^{n\times n}$ - диагональная матрица (ее диагональные элементы равны d_{ii}, а внедиагональные - нулю);

$E_n \in \mathbb{R}^{n\times n}$ - единичная матрица, $E_n = \operatorname{diag}(1, \dots, 1)$ (если из контекста очевиден порядок единичной матрицы, то матрица обозначается без индекса - E);

$\lambda_1(B), \dots, \lambda_n(B)$ - *собственные числа* матрицы $B \in \mathbb{R}^{n\times n}$, т.е. корни характеристического уравнения

$$\det(B - \lambda E_n) = 0;$$

$$\lambda_{\min}(B) = \min_{1\le i\le n} \lambda_i(B), \qquad \lambda_{\max}(B) = \max_{1\le i\le n} \lambda_i(B);$$

$$\lambda_i = \lambda_i(A), \quad i = 1, \dots, n, \quad A \in \mathbb{R}^{n\times n}.$$

Квадратная матрица $U = (u_{ij}) \in \mathbb{R}^{n\times n}$ называется *верхней треугольной*, если все ее элементы ниже главной диагонали равны нулю, т.е. $u_{ij} = 0 \; \forall\, i > j$ (соответственно, квадратная матрица $L = (l_{ij}) \in \mathbb{R}^{n\times n}$ называется *нижней треугольной*, если все ее элементы ниже главной диагонали равны нулю, т.е. $l_{ij} = 0 \; \forall\, i < j$).

Следом квадратной матрицы $A \in \mathbb{R}^{n\times n}$ называется сумма ее диагональных элементов $\operatorname{tr} A = \sum_{i=1}^n a_{ii}$.

Матрица $A \in \mathbb{R}^{n\times n}$ называется *невырожденной*, если $\det A \neq 0$.

Обратной для невырожденной матрицы $A \in \mathbb{R}^{n\times n}$ называется такая матрица $A^{-1} \in \mathbb{R}^{n\times n}$, что $A^{-1}A = AA^{-1} = E_n$.

Квадратная матрица A называется *симметричной*, если $A = A^\top$. Квадратная матрица A называется *эрмитовой*, если $A = A^*$, где A^* – матрица эрмитово сопряженная к матрице A, т.е. $A^* = \bar{A}^\top$.

Симметричная матрица $A \in \mathbb{R}^{n\times n}$ называется *положительно (неотрицательно)* определенной, если $\forall a \in \mathbb{R}^n \backslash \{0\}$ выполнено $a^\top Aa > 0$ (соответственно, $a^\top Aa \geq 0$).

$\mathbb{R}_n^{\geq}$ - множество неотрицательно определенных квадратных матриц порядка n.

$\mathbb{R}_n^{>}$ - множество положительно определенных квадратных матриц порядка n.

Для любых матриц $A, B \in \mathbb{R}_n^{\geq}$ запись $A > B$ ($A \geq B$) означает, что $A - B \in \mathbb{R}_n^{>}$ (соответственно, $A - B \in \mathbb{R}_n^{\geq}$).

Матрицы называются *согласованными* относительно некоторой операции, если эта операция определена.

1.2.2. Симметричные, положительно и неотрицательно определенные матрицы.

Теорема 1.1 (теорема о спектральном разложении). *Если* $A = A^\top \in \mathbb{R}^{n\times n}$, *то выполнено*

$$P^\top AP = \Lambda, \qquad A = P\Lambda P^\top. \tag{1.1}$$

где P *– ортогональная* $n \times n$*-матрица (т.е.* $P^\top P = PP^\top = E_n$*), столбцами которой являются ортонормированные собственные векторы матрицы* A, $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, *где* λ_i *– собственные числа матрицы* A.

Доказательство имеется в [4, 6, 14].

Теорема 1.2. *Матрица* $A \in \mathbb{R}^{n\times n}$ *неотрицательно определена тогда и только тогда, когда существует матрица* $F \in \mathbb{R}^{n\times n}$, *такая, что* $A = FF^\top$.

Теорема 1.3. *Матрица* $A \in \mathbb{R}^{n\times n}$ *положительно определена тогда и только тогда, когда существует невырожденная матрица* $F \in \mathbb{R}^{n\times n}$, *такая, что* $A = F^\top F$.

В теоремах 1.2 и 1.3 необходимость вытекает из теоремы 1.1, а достаточность – из определения неотрицательной (положительной) определенности.

Теорема 1.4. *Любую матрицу* $A \in \mathbb{R}_n^{>}$ *можно представить в виде* $A = FF^\top$, *где матрица* $F \in \mathbb{R}^{n\times n}$ *невырождена и является верхней треугольной.*

Доказательство имеется в [6].

Теорема 1.5. *Если* $A \in \mathbb{R}_n^>$, *то*

$$\lambda_i \geq 0, \quad a_{ii} \geq 0, \quad \det A = \prod_{i=1}^{n} \lambda_i,$$

$$a_{ij} \leq (a_{ii} + a_{jj})/2, \tag{1.2}$$

где $i, j = 1, \ldots, n$.

Если $A \in \mathbb{R}_n^>$, *то*

$$\lambda_i > 0, \quad a_{ii} > 0, \quad \det A > 0, \quad a_{ij} < (a_{ii} + a_{jj})/2, \quad i, j = 1, \ldots, n.$$

Теорема 1.6. *Если* $A \in \mathbb{R}_n^>$, $B \in \mathbb{R}_n^>$, $A + B \in \mathbb{R}_n^>$.

Теорема 1.7. *Если* $A \in \mathbb{R}_n^>$, *то* $A^{-1} \in \mathbb{R}_n^>$.

Теорема 1.8. *Пусть* $A \in \mathbb{R}_n^>$, $B \in \mathbb{R}^{n \times m}$, $\operatorname{rank} B = m \leq n$. *Тогда* $B^\top AB \in \mathbb{R}_n^>$. *В частности,* $B^\top B \in \mathbb{R}_n^>$.

Доказательство теорем 1.5 – 1.8 вытекают из определений неотрицательной и положительной определенности матриц и теорем 1.1 – 1.3. Комментария требует только вывод формулы 1.2. Эта формула следует из того, что

$$(e_i - e_j)^\top A(e_i - e_j) = e_i^\top Aa_i + e_j^\top Aa_j - 2e_i^\top Aa_j = a_{ii} + a_{jj} - 2a_{ij} \geq 0,$$

где $e_i \in \mathbb{R}^n$ – вектор, все компоненты которого равны нулю кроме i-й, равной единице.

Теорема 1.9. *Пусть матрица* $A \in \mathbb{R}_n^>$ *симметрична,* $\lambda_1 \geq \ldots \geq \lambda_n$ – *ее собственные числа и* $p_1, \ldots, p_n$ – *соответствующие им ортонормированные собственные векторы. Тогда*

$$\sup_{a \in \mathbb{R}^n \setminus \{0\}} \left\{ \frac{a^\top Aa}{a^\top a} \right\} = \lambda_1, \tag{1.3}$$

$$\inf_{a \in \mathbb{R}^n \setminus \{0\}} \left\{ \frac{a^\top Aa}{a^\top a} \right\} = \lambda_n, \tag{1.4}$$

причем экстремумы достигаются соответственно на p_1 *и* p_n.

Д о к а з а т е л ь с т в о. Формулу (1.1) перепишем в виде:

$$A = \sum_{i=1}^{n} \lambda_i p_i p_i^\top, \qquad \sum_{i=1}^{n} p_i p_i^\top = E_n.$$

В силу того, что $\{p_i\}_{i=1}^n$ – базис в $\mathbb{R}^n$ любой вектор $a \in \mathbb{R}^n$ представим в виде

$$a = \sum_{i=1}^{n} c_i p_i.$$

Поэтому

$$\frac{a^\top A a}{a^\top a} = \sum_{i=1}^{n} c_i^2 \lambda_i \left(\sum_{i=1}^{n} c_i^2 \right)^{-1}.$$

Очевидно, что супремум и инфимум этого выражения относительно векторов $(c_1, \ldots, c_n)^\top$ равны соответственно λ_1 и λ_n, причем супремум достигается при $a = p_1$, а инфимум – при $a = p_n$. □

1.2.3. След.

Теорема 1.10. *а) Если* $A \in \mathbb{R}^{n\times n}$, $c \in \mathbb{R}$, *то* $\operatorname{tr} A = \operatorname{tr} A^\top$, $\operatorname{tr} cA = c \operatorname{tr} A$;
б) если $A, B \in \mathbb{R}^{n\times n}$, *то*

$$\operatorname{tr}(A + B) = \operatorname{tr} A + \operatorname{tr} B; \tag{1.5}$$

в) если $A \in \mathbb{R}^{n\times m}$, $B \in \mathbb{R}^{m\times n}$, *то*

$$\operatorname{tr} AB = \operatorname{tr} BA; \tag{1.6}$$

г) если $a, b \in \mathbb{R}^n$, *то*

$$\operatorname{tr} ab^\top = \operatorname{tr} a^\top b;$$

д) если $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, *то*

$$\operatorname{tr}(Abb^\top) = \operatorname{tr}(bb^\top A) = b^\top A b;$$

е) если $A, P \in \mathbb{R}^{n\times n}$, $P^\top P = E_n$, *то*

$$\operatorname{tr} PAP^\top = \operatorname{tr} A. \tag{1.7}$$

Все утверждения теоремы легко следуют из определения операции trace (след).

Теорема 1.11. *Если A – симметричная матрица из $\mathbb{R}^{n\times n}$, то*

$$\operatorname{tr} A = \sum_{i=1}^{n} \lambda_i, \tag{1.8}$$

$$\operatorname{tr} A^s = \sum_{i=1}^{n} \lambda_i^s, \qquad s = -1, 0, 1, 2, \ldots.$$

Доказательство следует из теоремы 1.1 и (1.7) с учетом того, что при $PP^\top = E_n$ выполнено $(P^\top AP)^s = P^\top A^s P$.

Теорема 1.12. *Пусть A – симметричная $n \times n$-матрица. Необходимым и достаточным условием ее неотрицательной определенности является выполнение для всех $B \in \mathbb{R}_n^{\geq}$ неравенства* $\operatorname{tr} AB \geq 0$.

Д о к а з а т е л ь с т в о. По теореме (1.1) имеем

$$A = P\Lambda P = \sum_{i=1}^{n} \lambda p_i p_i^\top,$$

где P_i – ортонормированные собственные векторы матрицы A, соответствующие собственным числам λ_i. Отсюда следует:

$$\operatorname{tr} AB = \operatorname{tr} \sum_{i=1}^{n} \lambda_i p_i p_i^\top B = \sum_{i=1}^{n} \lambda_i p_i^\top B p_i.$$

Поскольку $B \in \mathbb{R}_n^{\geq}$, величины $p_i^\top B p_i$ $(i = 1, \ldots, n)$ неотрицательны.

Если $A \in \mathbb{R}_n^{\geq}$, то по теореме 1.5 $\lambda_i \geq 0$, поэтому $\operatorname{tr} AB \geq 0$. С другой стороны, если $\operatorname{tr} AB \geq 0$ для всех $B \geq 0$, то это справедливо и для матрицы $B = p_i p_i^\top$. Следовательно,

$$\operatorname{tr} A p_i p_i^\top = \operatorname{tr} \left(\sum_{j=1}^{n} \lambda_j p_j p_j^\top \right) p_i p_i^\top = \lambda_i \geq 0.$$

Отсюда с учетом теоремы 1.1 следует, что $A \geq 0$. □

1.2.4. Ранг.

Теорема 1.13. *Для любых согласованных матриц A, B выполнено*

а) $\operatorname{rank} AB \leq \min(\operatorname{rank} A, \operatorname{rank} B)$;

б) $\operatorname{rank}(A + B) \leq \operatorname{rank} A + \operatorname{rank} B$.

Д о к а з а т е л ь с т в о. а) Столбцы матрицы AB являются линейными комбинациями столбцов матрицы A, поэтому число линейно независимых столбцов в матрице AB не больше, чем в матрице A. Следовательно, $\operatorname{rank} AB \le \operatorname{rank} A$. Аналогично $\operatorname{rank} AB \le \operatorname{rank} B$.

б) Пусть матрицы A и B имеют размер $p \times q$. Обозначим через $a_1, \ldots, a_q$ и $b_1, \ldots, b_q$ столбцы матриц A и B соответственно, и пусть

$$D = (A, B) = (a_1, \ldots, a_q, b_1, \ldots, b_q)$$

есть блочная матрица, составленная из матриц A и B.

Запишем матрицу $A + B$ в виде

$$A + B = (a_1 + b_1, \ldots, a_q + b_q).$$

Поскольку размерность пространства, порожденного набором векторов $(a_1, \ldots, a_q, b_1, \ldots, b_q)$ не меньше, чем размерность пространства для векторов $(a_1 + b_1, \ldots, a_q + b_q)$, то $\operatorname{rank}(A + B) \le \operatorname{rank} D$.

Покажем теперь, что $\operatorname{rank} D \le \operatorname{rank} A + \operatorname{rank} B$. Для этого удалим из набора $(a_1, \ldots, a_q, b_1, \ldots, b_q)$ все векторы b_i, линейно зависящие от векторов a_j $(j = 1, \ldots, q)$. Матрицу, составленную из оставшихся векторов b_i , обозначим через B_*. Имеем:

$$\begin{aligned} \operatorname{rank} D &= \operatorname{rank} A + \operatorname{rank} B_*, \\ \operatorname{rank} B_* &\le \operatorname{rank} B, \end{aligned}$$

откуда и вытекает требуемое. □

Теорема 1.14. *Пусть* $A \in \mathbb{R}^{n\times m}$, $B \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{m\times m}$, $\det B \ne 0$, $\det C \ne 0$. *Тогда*

$$\operatorname{rank} BAC = \operatorname{rank} A.$$

Д о к а з а т е л ь с т в о. В силу предыдущей теоремы

$$\operatorname{rank} A \ge \operatorname{rank} AC \ge \operatorname{rank} ACC^{-1} = \operatorname{rank} A.$$

Поэтому $\operatorname{rank} A = \operatorname{rank} AC$. Аналогично $\operatorname{rank} A = \operatorname{rank} BAC$. □

Теорема 1.15. *Если матрица симметрична, то ее ранг равен числу ненулевых ее собственных значений.*

Доказательство следует из теорем 1.1 и 1.14.

1.3. Нормы векторов и матриц

1.3.1. Векторные нормы. Пусть V – линейное пространство, вещественное или комплексное. В дальнейшем под линейным пространством V будем понимать $\mathbb{R}^n$ или $\mathbb{C}^n$. *Нормой* в линейном пространстве V называется отображение $\|\cdot\| : V \to \mathbb{R}$, ставящее в соответствие каждому вектору $x \in V$ число $\|x\| \in \mathbb{R}$ и удовлетворяющее аксиомам: $\forall\, x, y \in V$, $\alpha \in \mathbb{R}(\mathbb{C})$

1) $\|x\| \geq 0, \quad \|x\| = 0 \Leftrightarrow x = 0$ (неотрицательность),

2) $\|\alpha x\| = |\alpha| \cdot \|x\|$ (однородность),

3) $\|x + y\| \leq \|x\| + \|y\|$ (неравенство треугольника).

Линейное пространство V с заданной на нем нормой $\|\cdot\|$ называется *линейным нормированным пространством*. Число $\|x\|$ называется *нормой вектора* x.

Наиболее употребительными в арифметических пространствах являются:

1. *Октаэдрическая норма вектора* (1-норма)

$$\|x\|_1 = \sum_{k=1}^{n} |x_k|.$$

2. *Евклидова или сферическая норма вектора* (2-норма)

$$\|x\|_2 = \|x\|_E = \sqrt{\sum_{k=1}^{n} |x_k|^2}.$$

3. *Кубическая норма вектора* (∞-норма)

$$\|x\|_\infty = \max_{1 \leq k \leq n} |x_k|.$$

4. *Норма Гёльдера* (p-норма)

$$\|x\|_p = \left(\sum_{k=1}^{n} |x_k|^p \right)^{1/p}, \qquad 1 \leq p \leq \infty.$$

Нетрудно заметить, что первые три векторные нормы являются частным случаем нормы Гёльдера, соответственно для $p = 1, 2, \infty$.

1.3.2. Эквивалентность норм в конечномерном пространстве. Две нормы $\|x\|_1$ и $\|x\|_2$ в линейном пространстве V называются

эквивалентными, если существуют такие числа $c_1 > 0$, $c_2 > 0$, что для любого вектора $x \in V$ выполняются неравенства

$$\|x\|_1 \le c_1\|x\|_2 \quad \text{и} \quad \|x\|_2 \le c_2\|x\|_1.$$

Теорема 1.16. *В конечномерном пространстве любые две нормы эквивалентны.*

Доказательство имеется в [10].

1.3.3. Нормы матриц. Под *нормой матрицы* A с действительными или комплексными элементами понимают действительное число $\|A\|$, т.е. отображение $\|A\| : \mathbb{R}^{m\times n}(\mathbb{C}^{m\times n}) \to \mathbb{R}$, и удовлетворяющее аксиомам:

1) $\|A\| \ge 0, \quad \|A\| = 0 \Leftrightarrow A = 0$ (неотрицательность),

2) $\|\alpha A\| = |\alpha| \cdot \|A\|$ (однородность),

3) $\|A + B\| \le \|A\| + \|B\|$ (неравенство треугольника),

4) $\|AB\| \le \|A\| \cdot \|B\|$ (мультипликативность)

$\forall\, A, B \in \mathbb{R}^{m\times n}(\mathbb{C}^{m\times n})$ (согласованных относительно указанных операций матриц) и $\forall\, \alpha \in \mathbb{R}(\mathbb{C})$.

Иногда в определении нормы матрицы ограничиваются лишь первыми тремя аксиомами. В таком случае норму матрицы называют *обобщенной*.

Примерами матричных норм матрицы $A = (a_{ij})$ являются:

1) $\|A\| = \sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|$;

2) $\|A\|_1 = \max_{1\le j\le n}\sum_{i=1}^{m}|a_{ij}|$ – *максимально столбцовая норма*;

3) $\|A\|_\infty = \max_{1\le i\le m}\sum_{j=1}^{n}|a_{ij}|$ – *максимально строковая норма*;

4) $\|A\|_E = \left(\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2\right)^{1/2}$.

Норма $\|A\|_E$ называется *евклидовой* (*сферической, Фробениуса, Шура*) матричной нормой.

Для матрицы

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}$$

все эти нормы будут иметь соответственно значения:

1. $\|A\| = \sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}| = 1 + 2 + \ldots + 12 = 78.$

2. $\|A\|_1 = \max_{1\le j\le n}\sum_{i=1}^{m}|a_{ij}| = \max(1+4+7+10, 2+5+8+11, 3+6+9+12) = \max(22, 26, 30) = 30.$

3. $\|A\|_\infty = \max_{1\le i\le m}\sum_{j=1}^{n}|a_{ij}| = \max(1+2+3, 4+5+6, 7+8+9, 10+11+12) = \max(6, 15, 24, 33) = 33.$

4. $\|A\|_E = \left(\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2\right)^{1/2} = \sqrt{1^2+2^2+\ldots+12^2} = \sqrt{650}.$

Норму матрицы $A \in \mathbb{R}^{m\times n}$ ($\mathbb{C}^{m\times n}$) называют *согласованной с векторной нормой*, если для любого вектора $x \in \mathbb{R}^n(\mathbb{C}^n)$ выполняется условие

$$\|Ax\| \le \|A\| \cdot \|x\|.$$

Часто норму матрицы A вводят через нормы векторов, полагая

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

Такую норму матрицы называют *матричной нормой, подчиненной векторной норме*, или *матричной нормой, индуцированной векторной нормой.*

Нетрудно показазать, что любая матричная норма единичной матрицы E_n, подчиненная векторной норме равна 1. Из этого факта непосредственно следует, что сферическая (евклидова) матричная норма не подчинена никакой векторной норме.

Приведем примеры подчиненных матричных норм.

1. Для октаэдрической нормы вектора $\|x\|_1$, подчиненной нормой матрицы A является

$$\|A\|_1 = \max_{1\le j\le n}\sum_{i=1}^{m}|a_{ij}|.$$

2. Для евклидовой (сферической) нормы вектора $\|x\|_2$, подчиненной матричной нормой является *спектральная матричная норма*

$$\|A\|_2 = \sup_{x\neq 0}\frac{\|Ax\|_2}{\|x\|_2} = \sup_{\|x\|_2}\|Ax\|_2 = \sqrt{\max_{1\le i\le n}|\lambda_i|},$$

где $\lambda_i = \lambda_i(A^*A)$, $i = 1, 2, \ldots, n$.

3. Для кубической нормы вектора $\|x\|_\infty$, подчиненной матричной нормой является

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

Между различными матричными нормами устанавливаются определенные соотношения. Особенно много таких соотношений приведено в [20].

1.3.4. Сходимость по норме.

Утверждение 1.1. *В нормированном пространстве V отображение $\rho : V \times V \to \mathbb{R}$, определенное равенством*

$$\rho(x, y) = \|x - y\|, \qquad \forall\, x, y \in V,$$

является метрикой.

Аксиомы метрики непосредственно вытекают из аксиом нормы. □

Таким образом, в нормированном пространстве можно ввести расстояние между векторами и, значит, пользоваться предельным переходом. Последовательность векторов $\{x^{(k)}\}$ в нормированном пространстве V называется *сходящейся по норме* к вектору $a \in V$, если $\lim\limits_{k\to\infty} \|x^{(k)} - a\| = 0$, при этом вектор a называется *пределом последовательности* $\{x^{(k)}\}$ *по норме* $\|\cdot\|$.

О б о з н а ч е н и е: $\lim\limits_{k\to\infty} x^{(k)} = a$ или $x^{(k)} \to a$.

Утверждение 1.2. *Сходящаяся по норме последовательность имеет единственный предел.*

Д о к а з а т е л ь с т в о повторяет доказательство аналогичной теоремы для числовой последовательности [11] и основано на аксиоме треугольника: $\|a - b\| = \|a - x^{(k)} + x^{(k)} - b\| \le \|x^{(k)} - a\| + \|x^{(k)} - b\|$, где a и b – два предела последовательности $x^{(k)}$. □

Пусть $x_0 \in V$ и $r > 0$. Множество $S(x_0, r) = \{x \in V : \|x - x_0\| = r\}$ называется *сферой радиуса r с центром x_0 по норме* $\|\cdot\|$, а множество $B(x_0, r) = \{x \in V : \|x - x_0\| \le r\}$ – *замкнутым шаром радиуса r с центром x_0 по норме* $\|\cdot\|$.

В дальнейшем сферы и шары по евклидовой норме $\|\cdot\|_2$ будут обозначаться символами $S_E(x_0, r)$ и $B_E(x_0, r)$.

Утверждение 1.3. *Из любой последовательности векторов $x^{(k)} \in B_E(x_0, r)$ (или $S_E(x_0, r)$) можно выделить подпоследовательность, сходящуюся по норме $\|\cdot\|_2$ к вектору $a \in B_E(x_0, r)$ ($S_E(x_0, r)$ соответственно).*

Д о к а з а т е л ь с т в о. Без ограничения общности можно считать, что $x_0 = 0$. Доказательство проведем для сферы $S_E(r) = \{x \in V : \|x\|_2 = r\}$. Пусть $e_1, \ldots, e_n$ – ортонормированный базис пространства V и $x^{(k)} = \sum_{i=1}^n x_i^{(k)} e_i \in S_E(r)$. Тогда

$$\|x^{(k)}\|_2 = \left(\sum_{i=1}^n |x_i^{(k)}|^2 \right)^{1/2} = r.$$

Это означает ограниченность координат векторов $x^{(k)}$ рассматриваемой последовательности. Согласно теореме Больцано-Вейерштрасса [11] из этой последовательности можно выделить сходящуюся (покоординатно) подпоследовательность $\{x^{(k_m)}\}$. Пусть $x^{(k_m)}$ имеет координаты $x_1^{(k_m)}, \ldots, x_n^{(k_m)}$, сходящиеся соответственно к $a_1, \ldots, a_n$. Положим $a = \sum_{i=1}^n a_i e_i$. Тогда

$$\|x^{(k_m)} - a\|_2 = \left(\sum_{i=1}^n \left| x_i^{(k_m)} - a_i \right|^2 \right)^{1/2} \to 0,$$

следовательно, подпоследовательность $\{x^{(k_m)}\}$ сходится к вектору a по евклидовой норме.

Покажем, что $a \in S_E(r)$. Действительно, в очевидном неравенстве $|\|x\| - \|y\|| \le \|x - y\|$ положим $x = x^{(k_m)}$, $y = a$. Тогда $\left| \|x^{(k_m)}\|_2 - \|a\|_2 \right| \le \|x^{(k_m)} - a\|_2$, откуда следует, что

$$\|x^{(k_m)}\|_2 - \|x^{(k_m)} - a\|_2 \le \|a\|_2 \le \|x^{(k_m)}\|_2 + \|x^{(k_m)} - a\|_2$$

или, с учетом того, что $\|x^{(k_m)} - a\|_2 \to 0$,

$$\|x^{(k_m)}\|_2 - \varepsilon \le \|a\|_2 \le \|x^{(k_m)}\|_2 + \varepsilon, \quad \forall\, \varepsilon > 0,$$

если m достаточно велико. Следовательно, $\|a\|_2 = r$ и $a \in S_E(r)$. □

Пусть $\|\cdot\|_1$ и $\|\cdot\|_2$ две произвольные векторные нормы в пространстве V. Тогда из теоремы 1.16 об эквивалентности норм в конечномерном пространстве непосредственно следует, что *в конечномерном пространстве из сходимости по одной норме следует сходимость по любой другой норме*, так как

$$\|x^{(k)} - a\|_1 \le c_1 \|x^{(k)} - a\|_2.$$

1.4. Сингулярное разложение матриц

1.4.1. Сингулярные числа и векторы матриц. Возможность построения для симметричной (эрмитовой) матрицы канонического разложения с ортогональной (унитарной) трансформирующей матрицей позволяет для произвольной $(m \times n)$-матрицы получить аналог такого разложения. В дальнейшем все изложение ведется для матриц из $\mathbb{C}^{m\times n}$, поэтому все результаты справедливы также для матриц из $\mathbb{R}^{m\times n}$. При этом символ " $*$ " эрмитова сопряжения необходимо заменить на знак транспонирования "$\top$". Прежде всего заметим, что для произвольной матрицы $A \in \mathbb{C}^{m\times n}$ ранга r матрицы A^*A и AA^* являются симметричными (эрмитовыми) матрицами ранга r и порядков соответственно n и m. Причем они неотрицательны. Поэтому собственные числа таких матриц являются действительными неотрицательными числами.

Обозначим собственные числа матрицы A^*A через $\sigma_1^2, \sigma_1^2, \ldots, \sigma_n^2$ и будем считать, что $\sigma_1^2 \geq \sigma_2^2 \geq \ldots \geq \sigma_n^2$ ($\sigma_i \neq 0$ при $i = 1, \ldots, r$). Оператор с симметричной (эрмитовой) матрицей A^*A имеет ортонормированную систему собственных векторов $e_1, e_2, \ldots, e_n$ соответственно по $\sigma_1^2, \sigma_1^2, \ldots, \sigma_n^2$, т.е. таких векторов $e_1, e_2, \ldots, e_n$, что

$$A^*Ae_i = \sigma_i^2 e_i, \quad (e_i, e_j) = \begin{cases} 1, & \text{при } i = j; \\ 0, & \text{при } i \neq j,\ i, j = \overline{1, n}. \end{cases} \tag{1.9}$$

Эта система векторов переводится оператором с матрицей A в некоторую ортогональную систему векторов $Ae_1, Ae_2, \ldots, Ae_n$, так как

$$(Ae_i, Ae_j) = (A^*Ae_i, e_j) = \sigma_i^2(e_i, e_j) = 0 \quad \text{при} \quad i \neq j.$$

Кроме того, модуль вектора Ae_i равен σ_i, так как

$$|Ae_i| = \sqrt{(A^*Ae_i, e_i)} = \sqrt{\sigma_i^2(e_i, e_i)} = \sigma_i.$$

Поэтому вектор Ae_i отличен от нулевого вектора тогда и только тогда, когда $\sigma_i \neq 0$, т.е. при $i = \overline{1, r}$. Ненулевой вектор Ae_i является собственным вектором оператора AA^* по собственному значению σ_i^2, так как

$$AA^*(Ae_i) = A(A^*A)e_i = A(\sigma_i^2 e_i) = \sigma_i^2 Ae_i.$$

Верно и обратное. Следовательно, ненулевые характеристические числа матриц A^*A и AA^* совпадают с учетом их кратностей и их число равно r, а кратности нулевого собственного числа этих матриц равны

соответственно $n-r$ и $m-r$. Общих собственных чисел у матриц A^*A и AA^* будет $s=\min(m,n)$.

Арифметические значения $\sigma_1,\sigma_2,\ldots,\sigma_s$ ($\sigma_i\neq 0$ при $i=\overline{1,r}$) корней квадратных из общих собственных чисел матриц A^*A и AA^* называются *сингулярными* (или *главными*) *числами марицы* A.

В пространстве $\mathbb{C}^n$ примем за базис ортонормированную систему $e_1,e_2,\ldots,e_n$ собственных векторов оператора с матрицей A^*A и построим ортонормированную систему векторов

$$f_1=\frac{Ae_1}{|Ae_1|}=\frac{Ae_1}{\sigma_1},\ldots,f_r=\frac{Ae_r}{|Ae_r|}=\frac{Ae_r}{\sigma_r}.$$

Дополним эту систему любыми векторами $f_{r+1},\ldots,f_m$ до ортонормированного базиса в $\mathbb{C}^m$. По построению векторы $f_1,f_2,\ldots,f_m$ удовлетворяют соотношениям

$$Ae_i=\begin{cases}\sigma_i f_i, & \text{при } i\leq r,\\ 0, & \text{при } i>r.\end{cases}\tag{1.10}$$

Умножая эти равенства слева на A^* и учитывая, что $A^*Ae_i=\sigma_i^2e_i$, получим соотношения

$$A^*f_i=\begin{cases}\sigma_i e_i, & \text{при } i\leq r,\\ 0, & \text{при } i>r.\end{cases}\tag{1.11}$$

Ортонормированные базисы $e_1,e_2,\ldots,e_n$ и $f_1,f_2,\ldots,f_m$ пространств $\mathbb{C}^n$ и $\mathbb{C}^m$, связанные соотношениями (1.10) и (1.11), называют *сингулярными базисами*. Причем векторы $e_1,e_2,\ldots,e_n$ называют *правыми сингулярными векторами матрицы* A, а векторы $f_1,f_2,\ldots,f_m$ – ее *левыми сингулярными векторами*.

Оператор, имеющий в паре исходных базисов пространств $\mathbb{C}^n$ и $\mathbb{C}^m$ матрицу A, в сингулярных базисах $e_1,e_2,\ldots,e_n$ и $f_1,f_2,\ldots,f_m$ этих пространств, в силу определения матрицы оператора и соотношений (1.10), имеет $(m\times n)$-матрицу

$$\Sigma=\begin{pmatrix}\sigma_1 & & & 0\\ & \ddots & & \\ & & \sigma_r & \\ 0 & & & 0\end{pmatrix}.\tag{1.12}$$

При этом из формулы, устанавливающей связь между матрицами одного и того же оператора в разных базисах, получаем

$$A=Q\Sigma P^*,\tag{1.13}$$

где P – ортогональная (унитарная) матрица порядка n, столбцами которой служат столбцы координат векторов $e_1, e_2, \ldots, e_n$ в исходном базисе пространства $\mathbb{C}^n$, Q – ортогональная (унитарная) матрица порядка m, столбцами которой являются столбцы координат векторов $f_1, f_2, \ldots, f_m$ в исходном базисе пространства $\mathbb{C}^m$.

Разложение (1.13) называют *сингулярным разложением матрицы* A или сокращенно *SVD-разложением*, где SVD – сокращение (аббревиатура) английского термина "singular value decomposition".

Любая матрица из $\mathbb{C}^{m\times n}$ ($\mathbb{R}^{m\times n}$) обладает многими различными сингулярными разложениями. Это следует из некоторого произвола при построении векторов $e_1, e_2, \ldots, e_n$ и $f_1, f_2, \ldots, f_m$.

Сингулярному разложению (1.13) можно придать следующий вид:

$$A = U\Sigma_r V^*, \tag{1.14}$$

где

$$\Sigma_r = \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_r \end{pmatrix} - \text{ квадратная матрица порядка } r,$$

получающаяся из $(m \times m)$-матрицы Σ вычеркиванием $n - r$ нулевых столбцов справа и $m - r$ нулевых строк снизу, U – $(m \times r)$-матрица, состоящая из первых r столбцов матрицы Q, V^* – $(r \times n)$-матрица, состоящая из первых r строк матрицы P^*.

Разложение (1.14) называют *второй формой сингулярного разложения матрицы* A. В него входят матрицы меньших размерностей, чем в первую форму, и, кроме того, в нем матрица Σ_r – квадратная невырожденная. Все это может оказаться существенным, особенно при работе с сингулярным разложением на компьютере.

При $m \geq n$ сингулярному разложению (1.13) иногда придают вид

$$A = U\Sigma_n V^*, \tag{1.15}$$

где

$$\Sigma_n = \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_n \end{pmatrix}$$

– квадратная матрица, состоящая из первых n строк и столбцов матрицы Σ, U – $(m \times n)$-матрица, состоящая из первых n столбцов матрицы Q, $V^* = P^*$.

Если действительная (комплексная) матрица A симметричная (унитарная), то можно добиться, чтобы (см. [20]) в сингулярном разложении $A = Q\Sigma P^*$ ортогональные (унитарные) матрицы P и Q удовлетворяли условиям $Q = P$ и $P^* = P^\top$.

При конструировании сингулярного разложения на ЭВМ его обычно получают косвенным путем (см. [12]). Стандартную программу такого метода можно найти в пакете Matlab (см., например, [7]).

Сингулярное разложение находит самое широкое применение в теории и приложениях, которые будут рассмотрены позже: при вычислении псевдообратной матрицы, при отыскании псевдорешений систем линейных алгебраических уравнений (СЛАУ) и их проекций на пространства правых сингулярных векторов, при отыскании решений неустойчивых СЛАУ, при проведении сингулярного анализа модели выравнивающей функции по методу наименьших квадратов (МНК). В [7] приведен пример применения SVD-разложения для решения задачи сжатия изображений.

Пример 1.1. Вычислить сингулярные числа для матрицы

$$A = \begin{pmatrix} -1 & -7 \\ 1 & 7 \end{pmatrix}.$$

Р е ш е н и е. Для матрицы

$$A^*A = \begin{pmatrix} -1 & 1 \\ -7 & 7 \end{pmatrix} \begin{pmatrix} -1 & -7 \\ 1 & 7 \end{pmatrix} = \begin{pmatrix} 2 & 14 \\ 14 & 98 \end{pmatrix}$$

характеристический многочлен $|A^\top A - \lambda E| = \lambda(\lambda - 100)$ имеет корни $\lambda_1 = 100$, $\lambda_2 = 0$. Поэтому $\sigma_1 = \sqrt{\lambda_1} = 10$, $\sigma_2 = 0$.

Пример 1.2. Построить сингулярное разложение матрицы

$$A = \begin{pmatrix} 4 & -3i \\ -3i & 4 \end{pmatrix}.$$

Р е ш е н и е. Характеристический многочлен

$$|A^*A - \lambda E| = \begin{vmatrix} 25 - \lambda & 0 \\ 0 & 25 - \lambda \end{vmatrix} = (25 - \lambda)^2$$

матрицы A^*A имеет корни $\lambda_1 = \lambda_2 = 25$. Поэтому $\sigma_1 = \sigma_2 = \sqrt{25} = 5$. Следовательно, матрица Σ имеет вид

$$\Sigma = \begin{pmatrix} 5 & 0 \\ 0 & 5 \end{pmatrix}.$$

При $\lambda = 25$ система $(A^*A - \lambda E)v = 0$, т.е. система

$$\begin{cases} 0 \cdot v_1 + 0 \cdot v_2 &= 0, \\ 0 \cdot v_1 + 0 \cdot v_2 &= 0, \end{cases}$$

имеет фундаментальную систему решений, состоящую из двух решений, например, из решений $b_1 = (1,0)^\top$, $b_2 = (0,1)^\top$. Они уже ортонормированы, поэтому $e'_1 = (1,0)^\top$, $e'_2 = (0,1)^\top$. Из столбцов координат этих векторов построим матрицу

$$P = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Далее строим векторы

$$\begin{aligned} f_1 &= \frac{Ae_1}{\sigma_1} = \frac{1}{5}\begin{pmatrix} 4 & -3i \\ -3i & 4 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{5}\begin{pmatrix} 4 \\ -3i \end{pmatrix}, \\ f_2 &= \frac{Ae_2}{\sigma_2} = \frac{1}{5}\begin{pmatrix} 4 & -3i \\ -3i & 4 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{5}\begin{pmatrix} -3i \\ 4 \end{pmatrix}. \end{aligned}$$

Число этих ортонормированных векторов равно размерности пространства $\mathbb{C}^2$. Поэтому из столбцов их координат построим матрицу

$$Q = \begin{pmatrix} 4/5 & -3i/5 \\ -3i/5 & 4/5 \end{pmatrix}$$

и запишем искомое сингулярное разложение

$$A = Q\Sigma P^* = \begin{pmatrix} 4/5 & -3i/5 \\ -3i/5 & 4/5 \end{pmatrix}\begin{pmatrix} 5 & 0 \\ 0 & 5 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Глава 2

Нормальные решения и псевдорешения

2.1. Псевдорешения линейных систем

2.1.1. Нормальные решения. Рассмотрим произвольную СЛАУ общего вида:

$$Au = f, \qquad A \in \mathbb{C}^{m\times n}, \qquad f \in \mathbb{C}^m. \tag{2.1}$$

В системе (2.1) m обозначает число уравнений, а n – число неизвестных.

Если $m < n$, то система (2.1) называется *недоопределенной*, если $m = n$, то в системе (2.1) число уравнений равно числу неизвестных и ее называют системой с *квадратной матрицей* коэффициентов, а в случае $m > n$ система называется *переопределенной*. Последний случай наиболее часто встречается в задачах обработки экспериментальных данных.

Условия совместности, т.е. существования решения, системы (2.1) определяются известной теоремой Кронекера - Капелли (см., например, [10]):

$$\operatorname{rank}(A\vdots f) = \operatorname{rank}(A).$$

При этом возможны два различных варианта:

1. $\operatorname{rank}(A\vdots f) = \operatorname{rank}(A) = n$ – система (2.1) имеет *единственное* решение u_0 (в случае $m = n$ единственность решения эквивалентна условию $\det A \neq 0$);

2. $\operatorname{rank}(A\vdots f) = \operatorname{rank}(A) \neq n$ – система (2.1) имеет бесчисленное множество решений (*неединственность*) $U = \{u : Au = f\}$. В этом

случае вводится понятие *нормального* решения u_*, т.е. решения с минимальной евклидовой нормой $\|u\|_2$:

$$u_* = \operatorname*{argmin}_{u \in U} \|u\|_2.$$

Утверждение 2.1. *Нормальное решение u_* совместной системы линейных алгебраических уравнений, когда последняя имеет бесчисленное множество решений, определяется единственным образом.*

Доказательство. Если система $Au = f$ имеет неединственное решение, то множество всех ее решений $U \subset \mathbb{C}^n$ есть выпуклое множество в $\mathbb{C}^n$. В самом деле, пусть u_1, u_2 – два решения системы $Au = f$. Тогда $u = tu_1 + (1-t)u_2$ при $0 \le t \le 1$ будет тоже решением:

$$Au = tAu_1 + (1-t)Au_2 = tf + (1-t)f = f.$$

Здесь использована линейность $A : \mathbb{C}^n \to \mathbb{C}^m$. Далее, рассмотрим строго выпуклый функционал на $U \quad g(u) = \|u\|_2$, ограниченный снизу $g(u) \ge 0$. Всякая минимизирующая последовательность $u^k \in U$ будет ограниченной в $\mathbb{C}^n$, так как $\|u^1\|_2 > \|u^2\|_2 > \ldots > \|u^k\|_2 > \ldots$ и, следовательно, компактной в $\mathbb{C}^n$. Поэтому существует предел $u_* = \lim_{k\to\infty} u^k$ и в силу строгой выпуклости $g(u)$ предел единственный. □

2.1.2. Псевдорешения. В случае, когда $\operatorname{rank}(A\vdots f) > \operatorname{rank}(A)$, система (2.1) не имеет решений (несовместность). Тогда вводится понятие *псевдорешения* системы (2.1), под которым понимается решение системы

$$Au = f_{\text{пр}}, \tag{2.2}$$

где $f_{\text{пр}}$ есть проекция вектора f на $\operatorname{im} A$.

Утверждение 2.2. *Для любой матрицы $A \in \mathbb{C}^{m\times n}$ и вектора $f \in \mathbb{C}^m$ ортогональная проекция $f_{\text{пр}}$ вектора f на множество значений матрицы A, т.е. $\operatorname{im} A$, определяется единственным образом.*

Доказательство. Образ матрицы $\operatorname{im} A$ является подпространством $\mathbb{C}^m$ в силу линейности отображения $A : \mathbb{C}^n \to \mathbb{C}^m$. Пусть $e_1, \ldots, e_q$ – ортонормированный базис в $\operatorname{im} A$, $q \le m$. Тогда, очевидно, многочлен Фурье $f_{\text{пр}} = \sum\limits_{k=1}^{q} (f, e_k)e_k$ будет единственной ортогональной проекцией f на $\operatorname{im} A$. □

Из определения $f_{\text{пр}}$ также непосредственно следует, что

$$\operatorname{rank}(A\vdots f_{\text{пр}}) = \operatorname{rank} A,$$

т.е. система (2.2) всегда *совместна* и имеет единственное или бесконечное множество решений $U' = \{u : Au = f_{\text{пр}}\}$.

При этом также возможны два различных варианта:

1. $\operatorname{rank}(A\vdots f_{\text{пр}}) = \operatorname{rank}(A) = n$ – система (2.2) имеет *единственное* решение u_*, которое и называется *псевдорешением* системы (2.1);

2. $\operatorname{rank}(A\vdots f_{\text{пр}}) = \operatorname{rank}(A) \neq n$ – система (2.2) имеет бесчисленное множество решений (*неединственность*) $U' = \{u : Au = f_{\text{пр}}\}$. В этом случае вводится понятие *нормального псевдорешения* u_*, т.е. псевдорешения с минимальной евклидовой нормой $\|u\|_2$:

$$u_* = \operatorname*{argmin}_{u \in U'} \|u\|_2.$$

Иногда, в случае несовместности систем (2.1) (или (2.2)), среди бесчисленного множества решений (или псевдорешений) ищется решение (псевдорешение), ближайшее к некоторой заданной точке $u^0 = (u_1^0, \ldots, u_n^0)^\top$, называемой *пробным решением*. Такая точка может быть известна из каких-то априорных соображений (например, из физического смысла задачи); в противном случае полагают $u^0 = 0$.

Решение уравнения (2.1) (или (2.2)), ближайшее к пробному решению u^0, называется *нормальным относительно* u^0 *решением* (или соответственно *нормальным относительно* u^0 *псевдорешением*), т.е.

$$u_* = \operatorname*{argmin}_{u \in U} \|u - u^0\|_2 \quad \text{или} \quad u_* = \operatorname*{argmin}_{u \in U'} \|u - u^0\|_2.$$

Если ввести показатель несовместности системы (2.1)

$$\mu = \inf_{u \in \mathbb{C}^n} \|Au - f\|_2,$$

то определению решения (или нормального решения) отвечает $\mu = 0$, а псевдорешению (или нормальному псевдорешению) $\mu > 0$.

2.2. Линейная задача наименьших квадратов

Понятие псевдорешения линейной системы уравнений тесным образом связано с решением *линейной задачи наименьших квадратов.*

Множество решений в смысле наименьших квадратов системы (2.1) определяется как

$$U' = \{u \in \mathbb{C}^n : \|Au - f\|_2 = \min\} \tag{2.3}$$

и характеризуется следующей теоремой

Теорема 2.1. *Решение задачи (2.3) эквивалентно следующему условию ортогональности:*

$$u \in U' \;\Leftrightarrow\; A^*(f - Au) = 0.$$

Доказательство. Предположим, что u удовлетворяет условию $A^*r_u = 0$, где $r_u = f - Au$. Тогда для любого вектора $v \in \mathbb{C}^n$ $r_v = f - Av = r_u + A(u - v)$. Возводя в квадрат, получаем

$$\|r_v\|_2^2 = \|r_u\|_2^2 + 2(u - v)^* \underbrace{A^*r_u}_{=0} + \|A(u - v)\|_2^2 \geq \|r_u\|_2^2.$$

Теперь предположим, что $A^*r_u = z \neq 0$. Тогда, если $u - v = -\varepsilon z$,

$$\|v\|_2^2 = \|r_u\|_2^2 - 2\varepsilon\|z\|_2^2 + \varepsilon^2\|Az\|_2^2 < \|r_u\|_2^2$$

для достаточно малых ε. □

Вектор $r = f - Au$ обозначает невязку зависящую от u. Теорема 2.1 показывает, что невязка соответствующая решению задачи наименьших квадратов ортогональна подпространству $\operatorname{im} A$.

Таким образом, правая часть системы (2.1) (вектор f) есть декомпозиция двух ортогональных компонент

$$f = Au + r, \qquad r \perp Au,$$

где символ $\perp$ означает $(r, Au) = 0$.

Данная декомпозиция всегда единственна, даже, если решение u линейной задачи наименьших квадратов (2.3) не единственно.

Из теоремы 2.1 следует, что решение линейной задачи наименьших квадратов (2.1) удовлетворяет решению *системы нормальных уравнений (нормального уравнения)*

$$A^*Au = A^*f, \tag{2.4}$$

где матрица A^*A эрмитова и неотрицательно определена, а система (2.4) совместна.

Из теоремы 2.1 также непосредственно следует, что множество псевдорешений, определенных в 2.1.2 совпадает с множеством решений линейной задачи наименьших квадратов (2.3) и соответственно с множеством решений системы нормальных уравнений (2.4), т.е.

$$\begin{aligned} U' &= \{u \in \mathbb{C}^n : Au = f_{\text{пр}}\} = \{u \in \mathbb{C}^n : \|Au - f\|_2 = \min\} \\ &= \{u \in \mathbb{C}^n : A^*Au = A^*f\}. \end{aligned}$$

Теорема 2.2. *Матрица A^*A положительно определена тогда и только тогда, когда столбцы матрицы A линейно независимы.*

Доказательство. Если столбцы матрицы A линейно независимы, тогда из $u \neq 0 \Rightarrow Au \neq 0$ и поэтому

$$u \neq 0 \;\Rightarrow\; u^*A^*Au = \|Au\|_2^2 > 0.$$

Следовательно A^*A положительно определена.

С другой стороны, если столбцы линейно зависимы, тогда для некоторого $u_0 \neq 0$ должно выполняться $Au_0 = 0$ и $u_0^*A^*Au_0 = 0$ и поэтому A^*A не является положительно определенной. □

Замечание 2.2.1. К системе нормальных уравнений (2.4) можно прийти также, используя методы математического анализа, а именно, приравняв к нулю дифференциал

$$\begin{aligned} dF(u) &= du^* \cdot A^*Au + u^*A^*A \cdot du - f^*Adu - du^* \cdot A^*f = \\ &= du^* \cdot A^*Au + du^* \cdot A^*Au - du^* \cdot A^*f - du^* \cdot A^*f = \\ &= 2du^* \cdot (A^*Au - A^*f) \end{aligned}$$

функции

$$\begin{aligned} F(u) &= \|Au - f\|_2^2 = (Au - f)^*(Au - f) = (u^*A^* - f^*)(Au - f) = \\ &= u^*A^*Au - f^*Au - u^*A^*f - f^*f. \end{aligned}$$

Замечание 2.2.2. К системе нормальных уравнений (2.4) можно также формально прийти умножив слева на матрицу A^* левую и правую части системы (2.1). Такое преобразование называется *первой трансформацией Гаусса.*

Замечание 2.2.3. Систему нормальных уравнений (2.4) и уравнения определяющие вектор невязки можно скомбинировать в совместную систему из $(m+n)$ линейных уравнений

$$\begin{pmatrix} E_m & A \\ A^* & 0 \end{pmatrix} \begin{pmatrix} r \\ u \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}. \tag{2.5}$$

Матрица системы (2.5) квадратная и эрмитова (соответственно для $A \in \mathbb{R}^{m\times n}$ – симметричная), но знаконеопределенная, если $A \neq 0$. Расширенная система (2.5) используется для итерационного уточнения псевдорешений (см., например, [2]) и в некоторых методах, где матрица A является разреженной.

Из теоремы 2.2 следует, что если $\operatorname{rank}(A) = n$, тогда существует единственное псевдорешение, которое может быть записано в виде

$$u_* = (A^*A)^{-1}A^*f,$$

и соответствующий вектор невязки

$$r_* = f - Au_* = (E_m - P_A)f, \qquad P_A = A(A^*A)^{-1}A^*,$$

где P_A – ортогональный проектор (матрица ортогонального проектирования) на подпространство $\operatorname{im} A$.

Если $\operatorname{rank}(A) < n$, псевдорешение не единственно (бесконечное множество псевдорешений), однако нормальное псевдорешения определяется однозначно.

2.3. Псевдообращение

Если $A \in \mathbb{C}^{n\times n}$ и $\det A \neq 0$, то для нее существует обратная матрица A^{-1}. Если же $A \in \mathbb{C}^{m\times n}$ и $m \neq n$ или $A \in \mathbb{C}^{n\times n}$, но $\det A = 0$, то матрица A не имеет обратной и символ A^{-1} не имеет смысла. Однако как будет показано далее, для произвольной прямоугольной матрицы существует "псевдообратная" матрица A^+, которая обладает некоторым свойствами обратной матрицы и имеет важные применения при решении систем

линейных уравнений. В случае, когда $A \in \mathbb{C}^{n\times n}$ и $\det A \neq 0$, псевдообратная матрица A^+ совпадает с обратной A^{-1}. Приведенное в этом разделе определение псевдообратной матрицы было дано в 1920 г. Муром, указавшим на важные применения этого понятия. Позже независимо от Мура в несколько иной форме псевдообратная матрица определялась и исследовалась в работах Пенроуза (см., например, [6]).

2.3.1. Псевдообратная матрица. Рассмотрим матричное уравнение

$$AXA = A. \tag{2.6}$$

Если $A \in \mathbb{C}^{n\times n}$ и $\det A \neq 0$, то уравнение (2.6) имеет единственное решение $X = A^{-1}$. Если же A – произвольная прямоугольная $m \times n$-матрица, то искомое решение $X \in \mathbb{C}^{n\times m}$, но не определяется однозначно. В общем случае уравнение (2.6) имеет бесчисленное множество решений, которые называются *обобщенной обратной* матрицей и обозначаются $X = A^-$. Обобщенная обратная матрица A^- обладает тем свойством, что для любого вектора $b \in \mathbb{C}^m$, при котором система уравнений $Au = b$ совместна, вектор $u = A^-b$ является ее решением.

Среди этих решений имеется только одно, обладающее тем свойством, что его строки и столбцы являются линейными комбинациями соответственно строк и столбцов сопряженной матрицы A^+ [6].

Определение. Матрицу $A^+ \in \mathbb{C}^{n\times m}$ называют *псевдообратной* (или обобщенной обратной матрицей Мура-Пенроуза) к матрице $A \in \mathbb{C}^{m\times n}$, если

$$AA^+A = A, \qquad A^+ = UA^* = A^*V, \tag{2.7}$$

где U и V – некоторые матрицы.

Утверждение 2.3. *Матрица A^+, удовлетворяющая условиям (2.7), существует и единственна.*

Доказательство. Начнем с доказательства единственности. Пусть A_1^+ и A_2^+ – две различные псевдообратные матрицы. Тогда

$$AA_1^+A = A, \quad A_1^+ = U_1A^* = A^*V_1$$

и

$$AA_2^+A = A, \quad A_2^+ = U_2A^* = A^*V_2$$

с некоторыми матрицами U_1, V_1, U_2 и V_2. Положим $D = A_1^+ - A_2^+$, $U = U_1 - U_2$, $V = V_1 - V_2$. Тогда

$$ADA = 0, \quad D = UA^* = A^*V.$$

Но $D^* = V^*A$, поэтому

$$(DA)^*(DA) = A^*D^*DA = A^*V^*ADA = 0,$$

и, значит, $DA = 0$. Отсюда, используя формулу $D^* = AU^*$, находим, что

$$DD^* = DAU^* = 0.$$

Следовательно, $A_1^+ - A_2^+ = D = 0$.

Для доказательства существования матрицы A^+ предположим сначала, что $\operatorname{rank} A = n$ ($A \in \mathbb{C}^{m\times n}$ с $m \geq n$). Покажем, что в этом случае матрица

$$A^+ = (A^*A)^{-1}A^* \tag{2.8}$$

удовлетворяет условиям (2.7).

Свойство $AA^+A = A$ из (2.7), очевидно, выполнено, поскольку

$$AA^+A = A((A^*A)^{-1}((A^*A) = A,$$

где $A^*A \in \mathbb{C}_n^>$. Равенство $A^+ = UA^*$ выполнено с $U = (A^*A)^{-1}$. Равенство же $A^+ = A^*V$ выполняется, как легко проверить, если положить $V = A(A^*A)^{-2}A^*$.

Аналогичным образом показывается, что если $\operatorname{rank} A = m$ ($A \in \mathbb{C}^{m\times n}$ с $m \leq n$), то псевдообратной к матрице A является матрица

$$A^+ = A^*(AA^*)^{-1}. \tag{2.9}$$

Для доказательства существования псевдообратной матрицы в общем случае используем тот факт, что всякую матрицу $A \in \mathbb{C}^{m\times n}$ можно представить в виде произведения

$$A = B \cdot C \tag{2.10}$$

с матрицами $B \in \mathbb{C}^{m\times r}$ и $C \in \mathbb{C}^{r\times n}$, где $r = \operatorname{rank} A \leq \min(m, n)$.

Действительно, возьмем в качестве матрицы B матрицу, составленную из r независимых столбцов матрицы A. Тогда все столбцы матрицы A можно выразить через столбцы матрицы B, о чем и свидетельствует формула (2.10), задающая "скелетное" разложение матрицы A.

Положим теперь

$$A^+ = C^+B^+,$$

где согласно (2.8) и (2.9)

$$C^+ = C^*(CC^*)^{-1}, \quad B^+ = (B^*B)^{-1}B^*.$$

Тогда

$$AA^+A = BCC^*(CC^*)^{-1}(B^*B)^{-1}B^*BC = BC = A.$$

Далее, если положить $U = C^*(CC^*)^{-1}(B^*B)^{-1}C$, то легко проверить, что $UA^* = A^+$.

Аналогичным образом проверяется, что $A^+ = A^*V$ с

$$V = B(B^*B)^{-1}(CC^*)^{-1}(B^*B)^{-1}B. \qquad \square$$

Таким образом, для любой матрицы $A \in \mathbb{C}^{m\times n}$ псевдообратная матрица существует и единственная, причем для невырожденной квадратной матрицы A псевдообратная матрица $A^+ = A^{-1}$.

2.3.2. Характеризация Пенроуза. В своей работе 1955 г., которая, по всей вероятности, возродила интерес к обобщенному обращению, Пенроуз [21] характеризовал псевдообратную матрицу как (единственное) решение совокупности матричных уравнений. Псевдообратная матрица A^+, введенная выше, удовлетворяет условиям Пенроуза.

Утверждение 2.4. *Для любой матрицы $A \in \mathbb{C}^{m\times n}$ $X = A^+$, если и только если*

1. $AXA = A$;
2. $XAX = X$;
3. $(AX)^* = AX \quad$ и $\quad (XA)^* = XA$.

Доказательство имеется в [3], [1].

2.3.3. Псевдообращение и псевдорешения линейных систем.

Утверждение 2.5. *Псевдообратная матрица A^+ является наилучшим приближением (по методу наименьших квадратов) матричного уравнения*

$$AX = E_m. \tag{2.11}$$

Утверждение 2.5 можно также сформулировать в эквивалентном виде

Утверждение 2.6. *Псевдообратной матрицей для матрицы A является матрица $A^+ \in \mathbb{C}^{n\times m}$, столбцы которой нормальные псевдорешения системы линейных уравнений вида*

$$Ax = e_i, \qquad i = 1, \ldots, m, \tag{2.12}$$

где e_i – столбцы единичной матрицы E_m.

Доказательство см., например, в [6, 5, 18]. Это свойство псевдообратной матрицы, также может быть принято в качестве ее определения.

Утверждение 2.7. *Пусть $u_*^{(1)}$ и $u_*^{(2)}$ – нормальные псевдорешения двух систем линейных уравнений $Au = f^{(1)}$ и $Au = f^{(2)}$. Тогда $\beta u_*^{(1)} + \gamma u_*^{(2)}$ является нормальным псевдорешением системы $Au = \beta f^{(1)} + \gamma f^{(2)}$.*

Доказательство. Из $A^*Au_*^{(1)} = A^*f^{(1)}$ и $A^*Au_*^{(2)} = A^*f^{(2)}$ следует, что $\beta u_*^{(1)} + \gamma u_*^{(2)}$ удовлетворяет нормальной системе

$$A^*A(\beta u_*^{(1)} + \gamma u_*^{(2)}) = A^*(\beta f^{(1)} + \gamma f^{(2)}).$$

Далее, существуют столбцы $z^{(1)}$ и $z^{(2)}$ такие, что $u_*^{(1)} = A^*z^{(1)}$ и $u_*^{(2)} = A^*z^{(2)}$. Поэтому $\beta u_*^{(1)} + \gamma u_*^{(2)} = A^*(\beta z^{(1)} + \gamma z^{(2)}))$. □

Естественно, утверждение 2.7 может быть распространено на линейные комбинации произвольного числа столбцов.

Утверждение 2.8. *Псевдорешение системы линейных уравнений (2.1) может быть записано в виде $u_* = A^+f$.*

Действительно, столбец свободных членов f представляет собой линейную комбинацию столбцов матрицы E_m:

$$f = \beta_1 e_1 + \ldots + \beta_m e_m.$$

По определению псевдообратной матрицы и согласно утверждению 2.7 псевдорешение u_* есть линейная комбинация столбцов a_i^+ псевдообратной матрицы с теми же коэффициентами

$$u_* = \beta_1 a_1^+ + \ldots + \beta_m a_m^+.$$

Это равносильно доказываемому утверждению.

Псевдообратная матрица обладает следующим экстремальным свойством.

Утверждение 2.9. *Для любой матрицы $X \in \mathbb{C}^{n\times m}$ выполнено соотношение*

$$\|AA^+ - E_m\|_E \le \|AX - E_m\|_E.$$

При этом, если для какой-нибудь матрицы X, отличной от A^+, здесь имеет место равенство, то $\|A^+\|_E < \|X\|_E$.

Доказательство. По определению при любом i столбец a_i^+ псевдообратной матрицы дает минимальную невязку при подстановке в (2.12). Поэтому для i-го столбца матрицы X

$$\|Aa_i^+ - e_i\| \le \|Ax_i - e_i\|.$$

Если же тут при $x_i \ne a_i^+$ достигается равенство, то $\|a_i^+\| < \|x_i\|$. Заметим, что квадрат евклидовой нормы матрицы равен сумме квадратов ее столбцов. Следовательно, возводя в квадрат и суммируя приведенные соотношения по всем $i = 1, \ldots, m$, приходим к доказываемому утверждению. □

Утверждение 2.10. *Нормальное относительно u^0 псевдорешение системы (2.1) определяется формулой*

$$u_* = A^+ f + (E_n - A^+ A)u^0.$$

Доказательство. Согласно утверждению 2.8 столбец $A^+ f$ – нормальное псевдорешение и, следовательно, является частным решением системы (2.4). Остается доказать, что столбец $z = (E_n - A^+ A)u^0$ при произвольном u^0 – общее решение нормальной однородной системы $A^* Az = 0$. Докажем это.

Во-первых, для любого u^0

$$A^* A[(E_n - A^+)u^0] = A^* Au^0 - A^* AA^+ Au^0 = A^* Au^0 - A^* Au^0 = 0.$$

Это означает, что z – решение нормальной однородной системы.

Во-вторых, для любого решения z системы $A^* Az = 0$ найдется столбец u^0, при котором

$$z = (E_n - A^+ A)u^0.$$

В действительности можно просто положить $u^0 = z$, так как система $A^* Az = 0$ равносильна системе $Az = 0$, и потому

$$(E_n - A^* A)z = z - A^+ Az = z. \qquad \square$$

Замечание 2.3.1. Утверждения 2.8 и 2.10 имеют главным образом теоретическое значение, как и правило Крамера для невырожденных матриц. Нахождение псевдообратной матрицы не обязательно для вычисления нормального псевдорешения и требует больших вычислительных затрат.

2.3.4. Псевдообращение при помощи предельного перехода.

Теорема 2.3. *Имеют место соотношения*

$$\lim_{\lambda\to 0}(A^*A+\lambda^2E_n)^{-1}A^*=A^+ \tag{2.13}$$

и

$$\lim_{\lambda\to 0}A^*(AA^*+\lambda^2E_m)^{-1}=A^+. \tag{2.14}$$

Доказательство имеется в [1], [3].

Из теоремы 2.3 непосредственно получаем

Следствие 2.3.1. *Для матриц полного ранга* ($\operatorname{rank}A=\min(m,n)$) *имеют место соотношения*

$$A^+=\begin{cases}(A^*A)^{-1}A^*, & \text{если } \operatorname{rank}A=n;\\ A^*(AA^*)^{-1}, & \text{если } \operatorname{rank}A=m.\end{cases}$$

Рассмотрим систему линейных уравнений

$$(A^*A+\alpha E_n)u=A^*f,\qquad \alpha>0. \tag{2.15}$$

Обозначим ее решение через u_α. Тогда

$$u_\alpha=(A^*A+\alpha E_n)^{-1}A^*f \tag{2.16}$$

и следствие 2.3.1 показывают, что справедливо

Утверждение 2.11. *При $\alpha\to 0$ решение u_α системы (2.15) стремиться к нормальному псевдорешению системы $Au=f$.*

Это утверждение имеет существенное теоретическое и прикладное значение. Дело в том, что нормальное псевдорешение системы линейных уравнений не является непрерывной функцией от матрицы системы. Утверждение 2.11 показывает, что система может быть включена в семейство систем с параметром α таким образом, что решение системы непрерывно зависит от параметра. Этот результат получен с более общей точки зрения в теории регуляризирующих функцтоналов для некорректно поставленных задач (см. Тихонов и Арсенин [16]). Упомянутая теория в основном относится к уравнениям в бесконечномерных пространствах (например, интегральным и диффренциальным уравнениям в частных производных).

В конечномерном случае прямой необходимости во введении регуляризирующих функционалов нет. Однако отметим, что для СЛАУ роль регуляризирующего функционала может играть функция

$$F_\alpha(u, f, A) = \|f - Au\|_2^2 + \alpha^2 \|u\|_2^2$$

на арифметическом пространстве $\mathbb{C}^n$. Найдем значение u, при котором она достигает минимума. Иначе F_α можно записать так:

$$F_\alpha(u, f, A) = (f - Au)^*(f - Au) + \alpha u^* u.$$

Дифференцируя это выражение по u, находим

$$dF_\alpha(u, f, A) = -2du^* A^* f + 2du^* A^* Au + 2\alpha du^* u.$$

Дифференциал обращается в нуль для векторов u, удовлетворяющих системе уравнений, в точности совпадающей с (2.15). Как мы видели выше, детерминант матрицы системы отличен от нуля, и система имеет единственное решение (2.16) при любых f, A и $\alpha \neq 0$.

Обозначим это решение через u_λ, и пусть $F_\alpha(u, f, A) = \xi$. Если $\|u\| > \sqrt{\xi/\alpha}$, то $F_\alpha(u, f, A) > \xi$. Поэтому на сфере радиуса $\sqrt{\xi/\alpha}+1$ и вне ее F_α принимает значения, большие чем ξ. Если u_α не попало внутрь сферы, увеличим ее радиус до $\|u_\alpha\| + 1$. Так мы получим сферу, содержащую u_α и такую, что на ней и вне ее $F_\alpha(u, f, A) > \xi$. $F_\alpha(u, f, A) > \xi$. Функция непрерывна и внутри сферы имеет единственную стационарную точку. Поэтому эта точка является точкой минимума. Приведенные рассуждения показывают, что это будет абсолютный минимум.

Утверждение 2.11, по существу, утверждает, что при $\alpha \longrightarrow 0$ точка, где регуляризирующий функционал достигает минимума, стремиться к псевдорешению системы $Au = f$.

2.3.5. Свойства псевдообратной матрицы. Отметим следующие основные свойства псевдообратной матрицы:

1. $(A^*)^+ = (A^+)^*$;
2. $(A^+)^+ = A$;
3. $(AA^+)^2 = AA^+, \quad (A^+A)^2 = A^+A.$

Первое свойство означает, что операция перехода к сопряженной матрице и к псевдообратной матрице перестановочны между собой. Равенство 2 выражает собой взаимность понятия псевдообратной матрицы, так как согласно 2 псевдообратной матрицей для A^+ является исходная матрица A. Согласно равенствам 3 матрицы AA^+ и A^+A являются

эрмитовыми и *инволютивными* (квадрат каждой из этих матриц равен самой матрице).

Много других свойств псевдообратных матриц имеется в [1].

2.4. Вычисление псевдообратных матриц

В этом разделе приведены некоторые достаточно простые, но практически важные численные алгоритмы нахождения псевдообратных матриц. Рассмотрены три численных метода нахождения псевдообратных матриц: алгоритм Гревилля, итерационный метод Бен-Израэля и метод, использующий сингулярное разложение матрицы (SVD-разложение).

2.4.1. Метод Гревилля. Этот метод не требует вычисления детерминантов и может быть также использован для вычисления обратной матрицы A^{-1} (в случае, если $A \in \mathbb{C}^{n\times n}$ и $|A| \neq 0$). Метод Гревилля последовательного нахождения псевдообратной матрицы состоит в следующем. Пусть a_k – k-й столбец в матрице $A \in \mathbb{C}^{m\times n}$, $A_k = (a_1, \ldots, a_k) \in \mathbb{C}^{m\times k}$ – матрица, образованная первыми k столбцами матрицы A, b_k – последняя строка в матрице A_k^+ $(k = 1, \ldots, n,\ A_1 = a_1, A_n = A)$. Тогда

$$A_1^+ = a_1^+ = \begin{cases} (a_1^* a_1)^{-1} a_1^*, & \text{если } a_1 \neq 0, \\ 0, & \text{если } a_1 = 0, \end{cases}$$

и для $k > 1$ имеют место рекуррентные формулы

$$A_k^+ = \begin{pmatrix} B_k \\ b_k \end{pmatrix}, \quad B_k = A_{k-1}^+ - d_k b_k, \quad d_k = A_{k-1}^+ a_k,$$

$$b_k = c_k^+ = \begin{cases} (c_k^* c_k)^{-1} c_k^*, & \text{если } c_k \neq 0, \\ (1 + d_k^* d_k) d_k^* A_{k-1}^+, & \text{если } c_k = 0, \end{cases}$$

где $c_k = a_k - A_{k-1} d_k$, $A_{k-1} \in \mathbb{C}^{m\times(k-1)}$, $A_{k-1}^+ \in \mathbb{C}^{(k-1)\times m}$, $d_k \in \mathbb{C}^{k-1}$ и b_k – вектор-строка размерности m.

Матрица A_k^+, построенная по этим формулам, является псевдообратной к матрице A_k, $k = 1, 2, \ldots, n$. В частности, $A_n^+ = A^+$.

2.4.2. Метод Бен-Израэля. При вычислении псевдообратной матрицы A^+ к действительной матрице $A \in \mathbb{R}^{m\times n}$ можно пользоваться итерационной формулой Бен-Израэля

$$X^{(k+1)} = X^{(k)} \left[2E_m - AX^{(k)}\right], \quad X^{(0)} = \alpha A^\top, \quad k = 0, 1, 2, \ldots . \qquad (2.17)$$

Если α – число, удовлетворяющее условию

$$0 < \alpha < \frac{2}{\lambda_{\max}},$$

где $\lambda_{\max} = \lambda_{\max}(A^\top A) = \lambda_{\max}(AA^\top)$, то

$$\lim_{k\to 0} \|X^{(k)} - A^+\| = 0.$$

Обычно на практике полагают $\alpha = 1.6/\lambda_{\max}$, а так как $\lambda_{\max} = \|A\|_2^2 \leq \|A\|_1\|A\|_\infty$, то можно взять $\alpha = 1.6/\|A\|_1\|A\|_\infty$.

Для остановки итерационного алгоритма (2.17), можно воспользоваться критерием

$$\frac{\left|\|AX^{(k+1)} - E_m\|_E - \|AX^{(k)} - E_m\|_E\right|}{\|AX^{(k)} - E_m\|_E} \leq \delta, \tag{2.18}$$

где δ – заданное (достаточно малое) число. В этом случае $X^{(k)}$, удовлетворяющее (2.18), принимается за приближенное значение A^+. Это правило остановки итерационного алгоритма основано на экстремальном свойстве псевдообратной матрицы, сформулированном в утверждении 2.9.

Иногда, возможно использовать, более простое по числу арифметических операций, правило остановки:

$$\frac{\|X^{(k+1)} - X^{(k)}\|}{\|X^{(k)}\|} \leq \delta', \tag{2.19}$$

где $\|\cdot\|$ – какая-либо матричная норма, δ' – заданное малое число. В некоторых случаях, целесообразно требовать одновременного выполнение условий (2.18) и (2.19).

2.4.3. Метод основанный на сингулярном разложении матриц. Как показано в разделе 1.4 для любой матрицы $A \in \mathbb{C}^{m\times n}$ существует сингулярное разложение (SVD-разложение) вида (1.14). Тогда из утверждения 2.9 непосредственно следует, что псевдообратная матрица

$$A^+ = V\Sigma_r^+ U^*,$$

где $\Sigma_r^+ = \operatorname{diag}\left(\sigma_1^{-1}, \ldots, \sigma_r^{-1}\right)$.

Данный метод является самым надежным (по точности) способом определения псевдообратных матриц, но в тоже время он является самым трудоемким (с вычислительной точки зрения) методом.

2.5. Типовые примеры

Пример 2.5.1. Рассмотрим систему, состоящую из одного уравнения:

$$a_1u_1 + a_2u_2 = f \quad (a_1^2 + a_2^2 \neq 0,\ a_1, a_2, f \in \mathbb{R}). \tag{2.20}$$

Требуется найти нормальное решение этой системы.

Р е ш е н и е. Для нахождения нормального решения u_* системы (2.20) можно воспользоваться псевдообратной матрицей,

$$u_* = A^+ f,$$

где $A = (a_1, a_2) \in \mathbb{R}^{1\times 2}$, т.е. матрица A – полного строкового ранга.

Используя свойство псевдообратной матрицы (из следствия 2.3.1) имеем

$$A^+ = A^\top (AA^\top)^{-1} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \left[\begin{pmatrix} a_1 & a_2 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \right]^{-1} = \frac{1}{a_1^2 + a_2^2} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}.$$

Отсюда нормальное решение системы (2.20) равно

$$u_* = \left(\frac{a_1 f}{a_1^2 + a_2^2}, \frac{a_2 f}{a_1^2 + a_2^2} \right)^\top.$$

Пример 2.5.2. Пусть

$$\left(A = \begin{matrix} 1 & -1 & 0 \\ -1 & 2 & 1 \\ 2 & -3 & -1 \\ 0 & 1 & 1 \end{matrix} \right).$$

Вычислить псевдообратную матрицу A^+ с помощью метода Гревилля.

Р е ш е н и е.

$$A_1^+ = (A_1^\top A_1)^{-1} = 1/6 \cdot A_1^\top = (1/6,\ -1/6,\ 1/3,\ 0),$$

$$d_2 = A_1^+ a_2 = -3/2, \quad c_2 = a_2 - A_1 d_2 = \begin{pmatrix} 1/2 \\ 1/2 \\ 0 \\ 1 \end{pmatrix},$$

$$b_2 = c_2^+ = (c_2^\top c_2)^{-1} c_2^\top = (1/3,\ 1/3,\ 0,\ 2/3),$$

$$B_2 = A_1^+ - d_2 b_2 = (2/3,\ 1/3,\ 1/3,\ 1).$$

Таким образом,

$$A_2^+ = \begin{pmatrix} 2/3 & 1/3 & 1/3 & 1 \\ 1/3 & 1/3 & 0 & 2/3 \end{pmatrix}.$$

Далее

$$d_3 = A_2^+ a_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{и} \quad c_3 = a_3 - A_2 d_3 = 0.$$

Поэтому

$$b_3 = (1 + d_3^\top d_3)^{-1} d_3^\top A_2^+ = (1/3,\ 1/3) A_2^+ = (1/3,\ 2/9,\ 1/9,\ 5/9)$$

и

$$B_3 = A_2^+ - d_3 b_3 = \begin{pmatrix} 2/3 & 1/3 & 1/3 & 1 \\ 1/3 & 1/3 & 0 & 2/3 \end{pmatrix} -$$

$$- \begin{pmatrix} 1/3 & 2/9 & 1/9 & 5/9 \\ 1/3 & 2/9 & 1/9 & 5/9 \end{pmatrix} = \begin{pmatrix} 1/3 & 1/9 & 2/9 & 4/9 \\ 0 & 1/9 & -1/9 & 1/9 \end{pmatrix},$$

$$A^+ = A_3^+ = \begin{pmatrix} 1/3 & 1/9 & 2/9 & 4/9 \\ 0 & 1/9 & -1/9 & 1/9 \\ 1/3 & 2/9 & 1/9 & 5/9 \end{pmatrix}.$$

Пример 2.5.3. Найти псевдообратную матрицу A^+ к матрице

$$A = \begin{pmatrix} 1 & i \\ -i & 1 \\ 1 & 1 \end{pmatrix}.$$

Р е ш е н и е. Ранг матрицы A равен числу ее столбцов. Поэтому применима формула из следствия 2.3.1. По ней получаем

$$A^+ = (A^* A)^{-1} A^* =$$

$$= \left(\begin{pmatrix} 1 & i & 1 \\ -i & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & i \\ -i & 1 \\ 1 & 1 \end{pmatrix} \right)^{-1} \begin{pmatrix} 1 & i & 1 \\ -1 & 1 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} 3 & 1+2i \\ 1-2i & 3 \end{pmatrix}^{-1} \begin{pmatrix} 1 & i & 1 \\ -i & 1 & 1 \end{pmatrix} = \frac{1}{4} \begin{pmatrix} 3 & -1-2i \\ -1+2i & 3 \end{pmatrix} \times$$

$$\times \begin{pmatrix} 1 & i & 1 \\ -i & 1 & 1 \end{pmatrix} = \frac{1}{4} \begin{pmatrix} 1+i & -1+i & 2-2i \\ -1-i & 1-i & 2+2i \end{pmatrix}.$$

Пример 2.5.4. Найти псевдообратную матрицу A^+ к матрице

$$A = \begin{pmatrix} 1 & -i & 1 \\ i & 1 & 1 \end{pmatrix}.$$

Р е ш е н и е. Ранг матрицы A равен числу ее строк. Поэтому применима формула из следствия 2.3.1. По ней получаем

$$A^+ = A^*(AA^*)^{-1} =$$

$$= \begin{pmatrix} 1 & i \\ -i & 1 \\ 1 & 1 \end{pmatrix} \left(\begin{pmatrix} 1 & i & 1 \\ -i & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & i \\ -i & 1 \\ 1 & 1 \end{pmatrix} \right)^{-1} =$$

$$= \begin{pmatrix} 1 & i \\ -i & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 3 & 1+2i \\ 1-2i & 3 \end{pmatrix}^{-1} = \frac{1}{4} \begin{pmatrix} 1 & i \\ -i & 1 \\ 1 & 1 \end{pmatrix} \times$$

$$\times \begin{pmatrix} 3 & -1-2i \\ -1+2i & 3 \end{pmatrix} = \frac{1}{4} \begin{pmatrix} 1+i & -1-i \\ -1+i & 1-i \\ 2-2i & 2+2i \end{pmatrix}.$$

Глава 3

Арифметика с плавающей точкой

Потребовалось не столь много времени после изобретения компьютера, чтобы прийти к общей точке зрения на правильный способ представления чисел в цифровой машине. Весь секрет в арифметике с плавающей точкой основан на математической записи действительного числа, реализованной на конкретном оборудовании. Перед тем как начать изучение точности алгоритмов вычислительной линейной алгебры, необходимо изучить эту главу.

3.1. Ограничения компьютерного представления действительных чисел

Так как цифровые компьютеры используют конечное число бит (разрядов) для представления действительных чисел, они могут представить только конечное подмножество действительных чисел (или комплексных чисел, вопрос о которых будет затронут в конце данной главы). Это ограничение представляет две сложности. Первая, представимые в компьютере числа не могут быть произвольно большими и малыми. Вторая, между представленными числами будут промежутки, т.е. между двумя машинными числами может не существовать ни одного машинного числа, в отличие от множества действительных чисел.

Современные компьютеры представляют числа достаточно большие и маленькие, так что первое из ограничений редко вызывает серьёзные проблемы. К примеру, широко используемый в компьютерах стандарт

IEEE-арифметики с двойной точностью позволяет представлять числа такие большие как 1.79×10^{308}, и такие маленькие как 2.23×10^{-308}, в итоге диапазон более достаточный для большинства прикладных вычислительных задач. Другими словами, переполнение сверху и снизу обычно совсем несерьезная проблема (но на нее иногда стоит обратить внимание, например, когда требуется вычислить определитель).

И совсем наоборот, проблема пустых промежутков между представляемыми в компьютере числами возникает повсюду в математических вычислениях. К примеру, в арифметике IEEE-стандарта с двойной точностью интервал $[1, 2]$ представляет собой дискретное множество

$$1, 1 + 2^{-52}, 1 + 2 \times 2^{-52}, 1 + 3 \times 2^{-52}, \ldots, 2. \tag{3.1}$$

Интервал $[2, 4]$ представляется теми же числами, только умноженными на 2:

$$2, 2 + 2^{-51}, 2 + 2 \times 2^{-51}, 2 + 3 \times 2^{-51}, \ldots, 4,$$

и в общем интервал $[2^j, 2^{j+1}]$ представляется посредством (3.1), умноженным на 2^j. Таким образом, в IEEE-арифметике с двойной точностью промежутки между соседними числами, в смысле относительных ошибок, никогда не превышают $2^{-52} \approx 2.22 \times 10^{-16}$. И кажется, что этим можно пренебречь, и это действительно так для большинства компьютерных вычислений, если используется *устойчивый компьютерный алгоритм*, определение которого будет рассмотрено в следующей главе. Но это может принести немало неожиданных хлопот в случае, если используемый компьютерный алгоритм спроектирован достаточно халатно.

3.2. Числа с плавающей точкой

IEEE-арифметика – пример арифметики, основанной на представлении действительных чисел с плавающей точкой. Это универсальный подход для большинства современных компьютеров. В системе чисел с плавающей точкой позиция десятичной или бинарной точки хранится отдельно от самих цифр, и промежутки между соседними представленными числами соответственно пропорциональны абсолютным величинам цифр. В этом принципиальное отличие представления вещественных чисел числами с плавающей точкой от представления с *фиксированной точкой*, где все промежутки одинаковы.

Рассмотрим определение идеализированной системы с плавающей точкой. Эта система состоит из дискретного подмножества $\mathbb{F}$ множества действительных чисел $\mathbb{R}$. Множество $\mathbb{F}$ чисел с плавающей точкой характеризуется четырьмя параметрами: *основанием системы счисления* $\beta \geq 2$, *разрядностью* p и *интервалом показателей* $[\nu^-, \nu^+]$. Каждое число $x \in \mathbb{F}$ представимо в виде

$$x = \pm \left(\frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \ldots + \frac{d_p}{\beta^p} \right) \beta^\nu, \tag{3.2}$$

где целые числа β, ν, $d_1, \ldots, d_p$ удовлетворяют неравенствам

$$0 \leq d_i \leq \beta - 1, \quad i = 1, \ldots, p; \quad \nu^- \leq \nu \leq \nu^+.$$

Часто d_i называют *разрядами*, p – *длиной мантиссы*, ν – *порядком числа* (или *експонентой*). *Мантиссой* (дробной частью) x называется число в скобках. Рассмотренное представление чисел, называемое представлением с *плавающей точкой*, используется в большинстве современных компьютеров. Основанием обычно является число $\beta = 2$ (с такими исключениями, как $\beta = 16$ для компьютеров IBM 370 и $\beta = 10$ для некоторых решающих таблиц и большинства калькуляторов). Например, $0.10101_2 \times 2^3 = 5.25_{10}$.

Число с плавающей точкой называется *нормализованным*, если старший разряд его мантиссы отличен от нуля ($d_1 \neq 0$). К примеру, число $0.10101_2 \times 2^3$ нормализовано, а число $0.010101_2 \times 2^4$ нет. Обычно числа с плавающей точкой нормализуют, что дает выигрыш в двух отношениях: всякое ненулевое число с плавающей точкой в этом случае имеет единственное представление в виде строки битов, и старший разряд двоичной мантиссы можно не хранить в явном виде (поскольку он всегда равен 1); за счет сэкономленного бита можно удлинить мантиссу.

3.3. Машинное эпсилон

Удобно считать, что *округление* – это отображение множества действительных чисел $\mathbb{R}$ в множество $\mathbb{F}$ чисел с плавающей точкой. Точность представления действительных чисел с использованием множества $\mathbb{F}$ традиционно определяется числом, которое называется *машинным эпсилон* ($\varepsilon_{\text{machine}}$) и зависит от разрядности p. При традиционном способе округления чисел имеем $\varepsilon_{\text{machine}} = \frac{1}{2}\beta^{1-p}$, при округлении отбрасыванием разрядов $\varepsilon_{\text{machine}} = \beta^{1-p}$. В общем случае, для арифметики с плава-

ющей точкой, имеющей p-разрядную мантиссу и основание β, *максимальная относительная ошибка представления* равна $0.5 \times \beta^{1-p}$. Это число – половина расстояния самого большого промежутка между полученными числами с плавающей точкой, т.е. $\varepsilon_{\text{machine}}$ имеет следующее свойство:

$$\forall\, x \in \mathbb{R},\ \exists\, x' \in \mathbb{F} \text{ такое, что } |x - x'| \leq \varepsilon_{\text{machine}}|x|. \tag{3.3}$$

Для значений β и p, стандартных для различных компьютеров, значение $\varepsilon_{\text{machine}}$ обычно лежит между 10^{-6} и 10^{-35}. В IEEE-арифметике с одинарной и двойной точностью $\varepsilon_{\text{machine}}$ принимает значения $2^{-24} \approx \approx 5.96 \times 10^{-8}$ и $2^{-53} \approx 1.11 \times 10^{-16}$ соответственно.

Область положительных нормализованных чисел IEEE-арифметики обычной точности простирается от 2^{-126} (*порог машинного нуля*) до $2^{127} \cdot (2 - 2^{-23}) \approx 2^{128}$ (*порог переполнения*) или приблизительно от 10^{-38} до 10^{38}. Границами области положительных нормализованных IEEE-чисел двойной точности являются 2^{-1022} (*порог машинного нуля*) и $2^{1023} \cdot (2 - 2^{-52}) \approx 2^{1024}$ (*порог переполнения*), т.е., приблизительно, 10^{-308} и 10^{308}.

Удобно считать, что *округление* – это некоторое отображение множества действительных чисел $\mathbb{R}$ в множество $\mathbb{F}$ чисел с плавающей точкой, т.е. $\text{fl} : \mathbb{R} \to \mathbb{F}$. Если $x \in \mathbb{R}$, а $\text{fl}(x) \in \mathbb{F}$, то в этих терминах неравенство (3.3) можно сформулировать в виде следующей аксиомы.

Аксиома 3.1. *Для всех* $x \in \mathbb{R}$ *существует* η, $|\eta| \leq \varepsilon_{\text{machine}}$, *такое, что*

$$\text{fl}(x) = x(1 + \eta). \tag{3.4}$$

Таким образом, относительная ошибка между действительным числом и его ближайшим приближением в системе с плавающей точкой всегда не превышает машинное эпсилон.

3.4. Арифметика чисел с плавающей точкой

Конечно, недостаточно просто представить действительные числа в цифровой машине, необходимо производить вычисления с ними. В компьютере все математические вычисления сводятся к некоторому числу элементарных арифметических операций, стандартный набор которых: $+$, $-$, $\times$ и $\div$. Математически эти символы определяют операции над полем $\mathbb{R}$. На компьютере они имеют точные аналоги операций над $\mathbb{F}$.

Соответствующие операции на множестве $\mathbb{F}$ чисел с плавающей точкой обозначаются как $\oplus$, $\ominus$, $\otimes$ и $\oslash$.

Для компьютерных вычислений справедливы следующие основные принципы. Пусть x и y произвольные числа с плавающей точкой, т.е. $x, y \in \mathbb{F}$. Пусть $*$ одна из операций $+$, $-$, $\times$ или $\div$, и пусть $\odot$ её аналог в арифметике с плавающей точкой. Тогда $x \odot y$ должно в точности давать

$$\mathrm{fl}(x * y) = x \odot y. \tag{3.5}$$

Если это свойство выполняется, тогда из (3.4) и (3.5) можно заключить, что компьютерные вычисления обладают простым и мощным свойством. Оно известно как *фундаментальная аксиома арифметики с плавающей точкой.*

Аксиома 3.2. *Для всех* $x, y \in \mathbb{R}$ *существует* η, $|\eta| \leq \varepsilon_{\text{machine}}$, *такое, что*

$$x \odot y = (x * y)(1 + \eta). \tag{3.6}$$

Другими словами, каждая операция в арифметике с плавающей точкой имеет относительную погрешность, не превышающую $\varepsilon_{\text{machine}}$.

3.5. Модификация машинного эпсилон

Анализ ошибок округления в этой книге основывается на (3.4) и (3.6) (аксиомах 3.1 и 3.2), но не на других деталях арифметики с плавающей точкой, описанных ранее. Это дает возможность применить рассмотренный анализ точности к компьютерным вычислениям, для которых не выполняется достаточно точно формула (3.5). Для любого компьютерного алгоритма (3.4) и (3.6) могут выполняться, если машинное эпсилон $\varepsilon_{\text{machine}}$ заменить несколько большим значением. К примеру, на компьютере, в котором действительные числа, принадлежащие промежуткам между числами с плавающей точкой, обрезаются, а не округляются, выражение (3.6) может выполняться, если машинное эпсилон $\varepsilon_{\text{machine}} = \beta^{1-p}$.

Самый простой способ предусмотреть такие проблемы, чтобы выполнялись (3.4) и (3.6), это модифицировать определение машинного эпсилон $\varepsilon_{\text{machine}}$.

С этого момента будем считать, что машинное эпсилон определяется не как в разделе 3.3, а определяется как наименьшее число, для которого (3.4) и (3.6) выполняются. Для большинства компьютеров, которые включают в себя реализацию IEEE-арифметики с плавающей точкой,

такое изменение определения машинного эпсилон не оказывает значительного влияния на его значение.

Тем не менее случается, что неожиданно большое значение машинного эпсилон может быть необходимо для того, чтобы (3.6) из аксиомы 3.2 выполнялось. В 1994 году процессор IntelPentium$^{\text{TM}}$ приобрел дурную славу, когда открылось, что по ошибке в таблице, используемой для реализации IEEE-арифметики с плавающей точкой двойной точности, его эффективная точность уменьшилась на 11 разрядов до машинного эпсилон $\varepsilon_{\text{machine}} \approx 6.1 \times 10^{-5}$. (Эта ошибка была вскоре устранена). На самом деле есть компьютеры, для которых (3.6) выполняется только для машинного эпсилон $\varepsilon_{\text{machine}} = 1$. К примеру, вычитание с плавающей точкой на компьютерах Cray, произведенных во второй половине 90-х годов, имело это свойство, так как операция вычитания была реализована без «сохранения цифр» (имеется в виду ее дробной части). Такие компьютеры не бесполезны, но они требуют анализа ошибок, отличного от того, который приведен в данной книге.

К счастью, положительные свойства аксиомы 3.2 и адаптация единообразных стандартов компьютерной арифметики стали широко распространенными среди производителей компьютеров в последние годы, и количество компьютеров, для которых не выполняется (3.6) с малыми значениями машинного эпсилон $\varepsilon_{\text{machine}}$, стремительно уменьшается. Действительно, IEEE-арифметика сама по себе быстро стала (начиная с 1996 года) стандартом для компьютеров любых классов, включая все IBM-совместимые персональные компьютеры и рабочие станции, произведенные SUN, DEC, Hewlett-Packard и IBM.

3.6. Комплексная арифметика с плавающей точкой

Комплексные числа с плавающей точкой обычно представляются как пара действительных чисел с плавающей точкой, а элементарные операции над ними вычисляются путем их приведения к операциям над действительной и мнимой частями. Результат аксиомы 3.2 верен для комплексных чисел аналогично тому, как верен и для действительных чисел, за исключением операций $\otimes$ и $\oslash$, для которых машинное эпсилон $\varepsilon_{\text{machine}}$ необходимо увеличить (как следует из $\varepsilon_{\text{machine}} = \frac{1}{2}\beta^{1-p}$) на множители порядка $2^{3/2}$ и $2^{5/2}$ соответственно. Определив однажды машинное эпсилон подобным образом, анализ ошибок округления для

комплексных чисел может быть проведен так же, как и для действительных чисел.

3.7. Упражнения

3.1. Сколко IEEE-чисел с двойной точностью содержится между парой ненулевых IEEE-чисел с обычной точностью ?

3.2. Верно ли, что всегда $\mathrm{fl}\left(\frac{a+b}{2}\right) \in [a, b]$?

3.3. Пусть отыскивается наименьший корень уравнения

$$y^2 - 140y + 1 = 0.$$

Вычисления производятся в десятичной системе счисления, причем в мантиссе числа после округления удерживаются 4 разряда. Какая из формул

$$y = 70 - \sqrt{4899} \qquad \text{или} \qquad y = \frac{1}{70 + \sqrt{4899}}$$

даёт более точный результат ?

3.4. Пусть вычисляется сумма

$$S_{1\,000\,000} = \sum_{j=1}^{1\,000\,000} \frac{1}{j^2}.$$

По какому алгоритму

$$S_0 = 0, \quad S_n = S_{n-1} + \frac{1}{n^2}, \quad n = 1, \ldots, 1\,000\,000$$

или

$$\sum_{1\,000\,000} = 0, \quad \sum_{n-1} = \sum_{n} + \frac{1}{n^2}, \quad n = 1\,000\,000, \ldots, 1$$

следует считать, чтобы суммарная вычислительная погрешность была меньше ?

3.5. Предложить наилучший способ вычисления знакопеременной суммы.

3.6. Система $\mathbb{F}$ чисел с плавающей точкой, определяемая как (3.2), включает много чисел, но не всё множество целых чисел.

(a) Дать точную формулу для наименьшего целого n, которое не принадлежит $\mathbb{F}$.

(b) В частности, какое значение n из (a) соответствует для IEEE-арифметики с обычной и двойной точностью ?

3.7. Пусть приближенное значение производной функции $f(x)$ определяется при $h \ll 1$ по одной из формул:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

или

$$f'(x) \approx \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h},$$

а сами значения $f(x)$ вычисляются с абсолютной погрешностью Δ. Какую погрешность можно ожидать при вычислении производной, если $|f^{(k)}| \le M_k$, $k = 0, 1, \ldots$?

3.8. Пусть вычисляется величина $S = a_1x_1 + \ldots + a_nx_n$, где коэффициенты a_i заданы с погрешностью δ. Найти погрешность вычисления S при условии, что $x_1^2 + \ldots + x_n^2 = 1$.

3.9. Пусть $|x| < 1$. В каком порядке лучше вычислять сумму $\sum\limits_{k=0}^{n} x^k$ с точки зрения уменьшения вычислительной погрешности ?

3.10. Пусть вычисления ведутся по формуле

$$y_{n+1} = 2y_n - y_{n-1} + h^2 f_n, \qquad n = 1, 2, \ldots,$$

y_0, y_1 заданы точно, $|f_n| \le M$, $h \ll 1$. Какую вычислительную погрешность можно ожидать при вычислении y^k ? Улучшится ли ситуация, если вычисления вести по формулам

$$\frac{y_{n+1} - y_n}{h} = f_n, \qquad \frac{y_n - y_{n-1}}{h} = z_n \ ?$$

3.11. Вычислить постоянную Эйлера

$$C = \lim_{n\to\infty} (1 + 1/2 + 1/3 + \ldots + 1/n - \ln n)$$

с 10 верными знаками.

Глава 4

Устойчивость компьютерных алгоритмов

Было бы просто замечательно, если бы численные алгоритмы могли обеспечить точные решения вычислительных задач. Так как в основном исходные вычислительные задачи непрерывны, в то время как цифровые компьютеры дискретны, это (точное решение исходной вычислительной задачи) в общем случае невозможно. Понятие "устойчивость" является стандартным способом описания возможности получения достаточно точного решения на основе компьютерного алгоритма в численном анализе.

4.1. Компьютерные алгоритмы

Математическая задача в общем случае может быть определена как функция (отображение) $g : X \to Y$ из нормированного векторного пространства данных X в нормированное векторное пространство решений Y.

Компьютерный алгоритм можно рассматривать как другое отображение $\tilde{g} : X \to Y$ между этими двумя пространствами. Уточним данное определение. Пусть даны фиксированные: задача g, компьютер (система вычислений с плавающей точкой которого удовлетворяет (3.6)) (но не обязательно (3.5)), алгоритм для решения g и реализация этого алгоритма в форме компьютерной программы. Пусть исходные данные $x \in X$ переведены в систему с плавающей точкой таким образом, что выполняется аксиома 3.1, и переданы на вход компьютерной программе. Результат – это числовой вектор с плавающей точкой, который принад-

лежит векторному пространству Y (так как алгоритм разработан для решения задачи g). Этот результат будет обозначаться – $\tilde{g}(x)$.

Ситуация вряд ли может быть хуже! Как минимум, $\tilde{g}(x)$ будет подвержена ошибкам округления. В зависимости от ситуации, результат может быть также подвержен другим видам ошибок и проблем, таким как плохая сходимость и даже помехам от других служб, работающих на компьютере, в случае, когда распределение вычислений между процессорами определяется в процессе самого расчета. Таким образом, "функция" $\tilde{g}(x)$ может даже принимать значения, различающиеся от одного запуска расчета к другому, т.е. может быть многозначной (на самом деле сама задача g, вполне возможно, допускает несколько решений, это разрешается в случае, когда неединственное решение допустимо, к примеру, любой из квадратных корней комплексного числа). Даже учитывая все эти осложнения, можно получить достаточно чёткие утверждения относительно $\tilde{g}(x)$, таким образом, относительно точности алгоритмов вычислительной линейной алгебры, основанные только на фундаментальных аксиомах 3.1 и3.2.

Обозначение с тильдой (˜) очень удобно. Так, если $\tilde{g}$ – компьютерный (вычислительный) аналог g, то другие, вычисленные на компьютере значения в этой книге, будут также часто обозначаться со знаком тильды. К примеру, вычисленное на компьютере решение системы уравнений $Au = f$ может быть обозначено как $\tilde{u}$.

4.2. Точность алгоритмов

Исключая тривиальные случаи, $\tilde{g}(x)$ не может быть непрерывной. Тем не менее хорошему алгоритму должно соответствовать хорошее приближение связанной с ним задачи. Для того чтобы выразить эту идею в цифрах, можно ввести *абсолютную ошибку* вычислений $\|\tilde{g}(x) - g(x)\|$ или *относительную ошибку*

$$\frac{\|\tilde{g}(x) - g(x)\|}{\|g(x)\|}. \tag{4.1}$$

В этой книге в основном используются относительные числа и таким образом, (4.1) будет основным стандартом измерения ошибок вычисления.

Если $\tilde{g}$ – "хороший" алгоритм, действительно можно ожидать, что относительная ошибка будет малой, т.е. порядка машинного эпсилон

$\varepsilon_{\text{machine}}$. Можно сказать, что алгоритм $\tilde{g}$ для задачи g *точен*, если для каждого $x \in X$ выполняется

$$\frac{\|\tilde{g}(x) - g(x)\|}{\|g(x)\|} = O(\varepsilon_{\text{machine}}). \tag{4.2}$$

Проще говоря, символ $O(\varepsilon_{\text{machine}})$ в (4.2) означает "порядка машинного эпсилон". Однако $O(\varepsilon_{\text{machine}})$ имеет точное определение, которое будет приведено позже. Также необходимо уточнить, как интерпретировать формулу (4.2) в случае, если знаменатель равен нулю.

4.3. Устойчивость

Если g плохо обусловлена, цель в достижении точности, определенной в (4.2), неоправданно завышена. Округление входных данных неизбежно для цифровых компьютеров, и даже если все последующие вычисления будут выполнены идеально точно, это единственное возмущение может привести к значительному изменению результата (вычисленного на компьютере решения). Вместо постановки цели точности алгоритма в таких случаях более подходит цель в обеспечении общей *устойчивости* компьютерного алгоритма.

Компьютерный алгоритм $\tilde{g}$ для задачи g называется *устойчивым*, если для каждого $x \in X$ выполняется

$$\frac{\|\tilde{g}(x) - g(\tilde{x})\|}{\|g(\tilde{x})\|} = O(\varepsilon_{\text{machine}}), \tag{4.3}$$

для некоторых $\tilde{x}$, удовлетворяющих

$$\frac{\|\tilde{x} - x\|}{\|x\|} = O(\varepsilon_{\text{machine}}). \tag{4.4}$$

Другими словами, устойчивый алгоритм дает ответ, близкий к верному, для задач с входными данными близкими к точным.

Почему принято именно такое определение устойчивости компьютерного алгоритма, станет ясным в следующей главе и дальнейших приложениях, рассмотренных в этой книге.

Хотя определение устойчивости, данное здесь, используется во многих разделах вычислительной линейной алгебры, условие $O(\varepsilon_{\text{machine}})$, вероятно, излишне жесткое для всех вычислительных задач в других разделах численного анализа. Например, таких как дифференциальные уравнения.

4.4. Обратная устойчивость

Многие алгоритмы вычислительной линейной алгебры удовлетворят условию, которое является как более сильным, так и более простым, чем рассмотренное выше условие устойчивости.

Компьютерный алгоритм $\tilde{g}$ для задачи g называется *обратно устойчивым*, если для каждого $x \in X$ выполняется:

$$\tilde{g}(x) = g(\tilde{x}) \quad \text{для некоторых} \quad \tilde{x} \quad \text{таких, что} \quad \frac{\|\tilde{x} - x\|}{\|x\|} = O(\varepsilon_{\text{machine}}). \tag{4.5}$$

Это определение устойчивости является более жестким по отношению к предыдущему определению устойчивости, так как $O(\varepsilon_{\text{machine}})$ в (4.3) было заменено нулем.

Другими словами, компьютерный алгоритм, обладающий обратной устойчивостью, дает точный результат (решение) для приближенной задачи с исходными данными близкими к точным.

Пример подобного алгоритма приводится в следующей главе.

4.5. Значение обозначения $O(\varepsilon_{\text{machine}})$

Поясним точное значение обозначения $O(\varepsilon_{\text{machine}})$ в формулах (4.2) – (4.5).

Обозначение

$$\varphi(t) = O(\psi(t)) \tag{4.6}$$

является стандартным в математике и имеет точное определение. Это выражение означает то, что существует положительная константа C, такая, что для всех t, стремящихся к предполагаемому пределу (т.е. $t \to 0$ или $t \to \infty$),

$$|\varphi(t)| \leq C\psi(t). \tag{4.7}$$

К примеру, выражение $\sin^2 t = O(t^2)$ при $t \to 0$ означает, что существует константа C такая, которая для всех достаточно малых t удовлетворяет $|\sin^2 t| \leq Ct^2$.

Также стандартом в математике являются выражения вида

$$\varphi(s,t) = O(\psi(t)) \quad \text{равномерно по} \quad s, \tag{4.8}$$

где φ – функция, которая зависит не только от t, но и также от другой переменной s. Слово "равномерно" показывает, что существует такая

константа C, как в (4.7), причем одна и та же для всех s. Таким образом, например,

$$(\sin^2 t)(\sin^2 s) = O(t^2)$$

выполняется равномерно для $t \to 0$, но равномерность теряется, если заменить $\sin^2 s$ на s^2.

В этой книге символ "O" используется, следуя этим стандартным определениям. Часто результат будет представляться в виде

$$\|\text{вычисленное значение (computed quantity)}\| = O(\varepsilon_{\text{machine}}). \tag{4.9}$$

Что же здесь означает (4.9) ? Во-первых, $\|$вычисленное значение$\|$, представляет норму некоторого числа или вектора, вычисленную компьютерным алгоритмом $\tilde{g}$ для задачи g, в зависимости от входных данных $x \in X$ для g и $O(\varepsilon_{\text{machine}})$. В качестве примера можно привести относительную ошибку (4.1). Во-вторых, неявно указанный процесс $O(\varepsilon_{\text{machine}}) \to 0$ (т.е. $\varepsilon_{\text{machine}}$ – переменная, зависящая от t в (4.8)). Третье, обозначение "O"применяется равномерно для всех входных данных $x \in X$ (т.е. переменная x соответствует s). В дальнейшем редко будет специально подчеркиваться равномерность по $x \in X$, но это всегда подразумевается.

В любой машинной арифметике, число $\varepsilon_{\text{machine}}$ – фиксированное значение. Говоря о пределе $O(\varepsilon_{\text{machine}}) \to 0$, полагаются на идеализацию компьютера или, точнее сказать, на идеализацию семейства компьютеров (так как речь идет о пределе). В итоге уравнение (4.9) означает, что если использовать компьютерный алгоритм решения задачи, на компьютерах, удовлетворяющих аксиомам 3.1 и 3.2, для последовательности значений $\varepsilon_{\text{machine}}$, стремящейся к нулю, тогда $\|$вычисленное значение$\|$ гарантированно уменьшается пропорционально $\varepsilon_{\text{machine}}$ или даже скорее. И для этих "идеальных" компьютеров требуется только лишь выполнение аксиом 3.1 и 3.2, и ничего более.

4.6. Зависимость от m и n, но не от A и f

В дальнейшем больше не будет обсуждаться значение $O(\varepsilon_{\text{machine}})$ в (4.2) – (4.5). Равномерность неявной константы в "O" может быть проиллюстрирована на следующем примере. Пусть рассматривается компьютерный алгоритм для решения невырожденной $m \times m$ системы уравнений $Au = f$ относительно u, и полагаем, что результат вычислений этого

алгоритма $\tilde{u}$ удовлетворяет

$$\frac{\|\tilde{u}-u\|}{\|u\|} = O(\kappa(A)\varepsilon_{\text{machine}}), \tag{4.10}$$

где $\kappa(A)$ – число обусловленности матрицы A, т.е. $\kappa(A) = \|A\|\|A^{-1}\|$.

Это предположение означает, что ограничение

$$\frac{\|\tilde{u}-u\|}{\|u\|} \leq C\kappa(A)\varepsilon_{\text{machine}} \tag{4.11}$$

сохраняется для единственной константы C, независимо от матрицы A или правой части f, для всех достаточно малых $\varepsilon_{\text{machine}}$.

Если знаменатель в формуле (4.11) равняется нулю, то выражение (4.11) определяется в виде

$$\|\tilde{u}-u\| \leq C\kappa(A)\varepsilon_{\text{machine}}\|u\|. \tag{4.12}$$

И здесь (в (4.12)) нет разницы с (4.10), т.к. если $\|u\| = 0$, то (4.12) поясняет значение выражения (4.10), а именно означает, что $\|\tilde{u}-u\| = 0$ для всех достаточно малых $\varepsilon_{\text{machine}}$.

Хотя константа C в выражениях (4.11) и (4.12) не зависит от A и f, в общем случае она зависит от m. Формально говоря, это следует из определения вычислительной задачи $g : X \to Y$. Если размерности, такие как m и n, определяющие задачу g, изменятся, то векторные пространства X и Y могут также измениться, что в результате приведет к новой задаче g'. Таким образом, на практике эффекты ошибок округления алгоритмов вычислительной линейной алгебры в общем случае растут с увеличением m и n. Однако этот рост обычно достаточно медленный, что не очень существенно на практике. Зависимость от m и n чаще всего линейная, квадратичная или кубическая (в самых плохих случаях), благодаря взаимоуничтожению ошибок (в связи с большим количеством арифметических операций).

В принципе, ограничение (4.9) может иметь зависимость от такого фактора (зависящего от размерности), как 2^m, что может, в свою очередь, сделать это ограничение неприменимым на практике. Например, подобная ситуация имеет место для метода исключения Гаусса, где имеются любопытные примеры, предупреждающие о том, что обсуждаемый вопрос требует внимательного к нему отношения. Тем не менее, как правило, когда в данной книге приведено выражение $O(\varepsilon_{\text{machine}})$, оно означает вероятность того, что ошибка решения в реальных вычислениях на реальном компьютере может превышать $\varepsilon_{\text{machine}}$ более чем в 100 или, возможно, даже 1000 раз.

4.7. Независимость от выбора векторных норм

Все определения, включающие $O(\varepsilon_{\text{machine}})$, умеют очень удобное свойство – они независимы от выбора векторной нормы, т.к. X и Y конечномерные пространства.

Теорема 4.1. *Для задачи g и компьютерного алгоритма $\tilde{g}$, определенных в конечномерных векторных пространствах X и Y, свойства точности, устойчивости и обратной устойчивости сохраняются независимо от выбора векторных норм в X и Y.*

Доказательство. Широко известно (и очень просто доказывается), что в конечномерных векторных пространствах все векторные нормы эквивалентны в смысле, что если $\|\cdot\|$ и $\|\cdot\|'$ – две нормы в одном и том же пространстве, то тогда существуют такие положительные константы C_1 и C_2, что $C_1\|x\| \le \|x\|' \le C_2\|x\|$ для всех x в этом пространстве. Отсюда следует, что замена нормы может повлиять на размер константы C, входящей в $O(\varepsilon_{\text{machine}})$, но никак на само существование такой константы. □

4.8. Упражнения

4.1. Проверить справедливость следующих выражений.

(a) $\sin x = O(1)$ при $x \to \infty$.

(b) $\sin x = O(1)$ при $x \to 0$.

(c) $\log x = O(x^{1/100})$ при $x \to \infty$.

(d) $n! = O((n/e)^n)$ при $n \to \infty$.

(e) $\text{fl}(\pi) - \pi = O(\varepsilon_{\text{machine}})$. (Здесь не предполагается, что $\varepsilon_{\text{machine}} \to 0$, а подразумевается, что для всех выражений $O(\varepsilon_{\text{machine}})$ в этой книге.)

(f) $\text{fl}(\pi) - \pi = O(\varepsilon_{\text{machine}})$, равномерно для всех целых n. (Здесь $n\pi$ соответствует точному математическому значению, а не результату, полученному вычислением в арифметике с плавающей точкой.)

4.2. (a) Показать, что

$$(1 + O(\varepsilon_{\text{machine}}))(1 + O(\varepsilon_{\text{machine}})) = 1 + O(\varepsilon_{\text{machine}}).$$

Точное значение этого утверждения состоит в том, что если g – функция удовлетворяющая

$$g(\varepsilon_{\text{machine}}) = (1 + O(\varepsilon_{\text{machine}}))(1 + O(\varepsilon_{\text{machine}}))$$

при $\varepsilon_{\text{machine}} \to 0$, тогда g также удовлетворяет $g(\varepsilon_{\text{machine}}) = 1+O(\varepsilon_{\text{machine}})$ при $\varepsilon_{\text{machine}} \to 0$.

(b) Показать, что $(1 + O(\varepsilon_{\text{machine}}))^{-1} = 1 + O(\varepsilon_{\text{machine}})$.

Глава 5

Обратный анализ ошибок компьютерных алгоритмов

В этой главе продолжается рассмотрение понятия устойчивости на конкретных примерах устойчивых и неустойчивых алгоритмов, а также обсуждается фундаментальная идея "обратного анализа ошибок связывающая обусловленность и устойчивость, мощь которой была доказана в теоретических исследованиях, начиная с 50-х годов прошлого века.

5.1. Устойчивость арифметики с плавающей точкой

Четыре простейших вычислительных задачи – это операции: $+$, $-$, $\times$ и $\div$. В них речь не идет о выборе алгоритма! Разумеется, для их реализации обычно используются операции с плавающей точкой $\oplus$, $\ominus$, $\otimes$ и $\oslash$, поддерживаемые компьютером. Оказывается, что в случае выполнения аксиом 3.1 и 3.2 эти четыре канонических примера обладают свойством обратной устойчивости.

Покажем это для операции вычитания, так как только от этой элементарной операции можно ждать высокоого риска неустойчивости. Рассмотрим пример, когда пространство исходных данных X – множество векторов в $\mathbb{C}^2$, а пространство решений Y – множество скаляров в $\mathbb{C}$. По теореме 4.1 в дальнейшем не нужно указывать конкретный тип норм в этих пространствах.

Для исходных данных $x = (x_1, x_2)^* \in X$, задача вычитания соответствует функции $g(x_1, x_2) = x_1 - x_2$ и рассматриваемый компьютерный

алгоритм может быть записан как

$$\tilde{g}(x_1, x_2) = \mathrm{fl}(x_1) \ominus \mathrm{fl}(x_2).$$

Это выражение означает, что вначале x_1 и x_2 переводятся в значения с плавающей точкой, а затем применяется операция $\ominus$. И теперь, на основании аксиомы 3.1, имеем

$$\mathrm{fl}(x_1) = x_1(1+\varepsilon_1), \qquad \mathrm{fl}(x_2) = x_2(1+\varepsilon_2)$$

для некоторых $|\varepsilon_1|, |\varepsilon_2| \le \varepsilon_{\text{machine}}$. Из аксиомы 3.2 следует, что

$$\mathrm{fl}(x_1) \ominus \mathrm{fl}(x_2) = (\mathrm{fl}(x_1) - \mathrm{fl}(x_2))(1+\varepsilon_3)$$

для некоторого $|\varepsilon_3| \le \varepsilon_{\text{machine}}$. Объединяя эти выражения, получаем

$$\begin{aligned} \mathrm{fl}(x_1) \ominus \mathrm{fl}(x_2) &= [x_1(1+\varepsilon_1) - x_2(1+\varepsilon_2)](1+\varepsilon_3) = \\ &= x_1(1+\varepsilon_1)(1+\varepsilon_3) - x_2(1+\varepsilon_2)(1+\varepsilon_3) = \\ &= x_1(1+\varepsilon_4) - x_2(1+\varepsilon_5) \end{aligned}$$

для некоторых $|\varepsilon_4|, |\varepsilon_5| \le 2\varepsilon_{\text{machine}} + O(\varepsilon^2_{\text{machine}})$ (см. упражнение 4.2). Другими словами, результат вычисления $\tilde{g}(x) = \mathrm{fl}(x_1) \ominus \mathrm{fl}(x_2)$ точно равен разности $\tilde{x}_1 - \tilde{x}_2$, где $\tilde{x}_1$ и $\tilde{x}_2$ удовлетворяют

$$\frac{|\tilde{x}_1 - x_1|}{|x_1|} = O(\varepsilon_{\text{machine}}), \qquad \frac{|\tilde{x}_2 - x_2|}{|x_2|} = O(\varepsilon_{\text{machine}}),$$

и любого $C > 2$ будет достаточно в качестве неявной константы в определении символа "O". Для любого выбора нормы $\|\cdot\|$ в пространстве $\mathbb{C}^2$ это приводит к (4.5).

5.2. Другие примеры

Пример 5.1. Скалярное произведение. Пусть даны векторы $x, y \in \mathbb{C}^m$ и требуется вычислить их скалярное произведение $\alpha = x^* y$. Банальный алгоритм – это вычисление попарных произведений $\overline{x_i} y_i$, посредством операции $\otimes$ и сложение их посредством $\oplus$. Можно показать, что этот алгоритм (вычисленный результат $\tilde{\alpha}$) обладает обратной устойчивостью.

Пример 5.2. Внешнее произведение векторов. С другой стороны, пусть требуется вычислить внешнее произведение векторов $A = xy^*$

для векторов $x \in \mathbb{C}^m$, $y \in \mathbb{C}^n$, имеющее ранг, равный 1. Банальный алгоритм – вычислить mn произведений $x_i\overline{y_i}$ посредством операции $\oplus$ и сложить их в результирующую матрицу $\tilde{A}$. Этот алгоритм устойчив, но не обладает обратной устойчивостью. Объясняется это тем, что результирующая матрица $\tilde{A}$ вряд ли будет иметь ранг, в точности равный 1, и таким образом, может быть в общем случае записана в виде $(x + \delta x)(y + \delta y)^*$. Как правило, задачи, в которых размерность пространства решений Y превышает размерность пространства исходных данных X задачи, очень редко обладают обратной устойчивостью.

Пример 5.3. Пусть используется операция $\otimes$ для вычисления $x+1$, где $x \in \mathbb{C}$, т.е. $\tilde{g}(x) = \mathrm{fl}(x) \oplus 1$. Этот алгоритм устойчив, но не обладает обратной устойчивостью. Причина в том, что при $x \approx 0$ операция сложения $\oplus$ вносит абсолютную ошибку порядка $O(\varepsilon_{\text{machine}})$. Относительно размера x эта ошибка не ограничена, и она не может быть объяснена малыми относительными возмущениями данных. Этот пример показывает, что обратная устойчивость – весьма специфическое свойство и является разумной целью далеко не во всех случаях. Если бы задача ставилась, как вычисление $x + y$ для данных x и y, тогда алгоритм мог бы обладать обратной устойчивостью.

Пример 5.4. Интересно, а что стоит ожидать от компьютерной программы или калькулятора, которые вычисляют $\sin x$ или $\cos x$? Ответ, опять же, стоит ожидать устойчивости, но не обратной устойчивости. Для $\cos x$ это следует из того, что $\cos 0 \neq 0$ аналогично предыдущему примеру. Как для $\sin x$, так и для $\cos x$ обратная устойчивость не получается из-за того, что функция имеет производную, равную нулю в нескольких точках. К примеру, пусть вычисляется $g(x) = \sin x$ на компьютере для $x = \pi/2 - \delta$, $0 < \delta \ll 1$. Пусть удалось получить при вычислениях абсолютно точный результат, округленный в системе с плавающей точкой: $\tilde{g}(x) = \mathrm{fl}(\sin x)$. Так как $g'(x) = \cos x \approx \delta$, то $\tilde{g}(x) = g(\tilde{x})$ для некоторого $\tilde{x}$, такого, что $\tilde{x} - x \approx (\tilde{g}(x) - g(x))/\delta = = O(\varepsilon_{\text{machine}}/\delta)$. Так как δ может быть сколь угодно малым, эта обратная ошибка вовсе не будет величиной порядка $O(\varepsilon_{\text{machine}})$.

5.3. Неустойчивый алгоритм

Это все были элементарные примеры. Вот один более солидный: использование характеристического многочлена для нахождения собственных значений матрицы.

Так как λ – собственное значение матрицы A, тогда и только когда $p(\lambda) = 0$, где $p(\lambda)$ – характеристический многочлен $\det(\lambda E - A)$, корни которого являются собственными значениями матрицы A. Этот факт дает метод вычисления собственных значений:

1. Найти коэффициенты характеристического многочлена.
2. Найти его корни.

Этот алгоритм не обладает не только обратной устойчивостью, но и вообще устойчивостью, и поэтому его не следует использовать в вычислительной практике. Даже когда задача нахождения собственных чисел хорошо обусловлена, этот алгоритм может выдавать ответы, которые имеют относительную ошибку во много раз большую, чем $\varepsilon_{\text{machine}}$.

Неустойчивость определяется из второго этапа метода вычисления собственных значений. Как хорошо известно, задача нахождения корней многочлена с заданными коэффициентами в общем случае плохо обусловлена. Из этого следует, что малые ошибки в коэффициентах характеристического многочлена приводят к значительному искажению корней, даже если сама задача нахождения корней выполнена с идеальной точностью.

К примеру, пусть $A = E$ – единичная матрица 2-го порядка. Собственные значения матрицы A слабочувствительны к возмущениям отдельных значений матрицы, и стабильный алгоритм должен бы вычислить их с ошибкой порядка $O(\varepsilon_{\text{machine}})$. Однако алгоритм, описанный выше, привносит ошибки порядка $\sqrt{\varepsilon_{\text{machine}}}$. Для того чтобы объяснить это, заметим, что характеристический многочлен имеет вид $\lambda^2 - 2\lambda + 1$. Когда вычисляются коэффициенты характеристического многочлена, могут ожидаться ошибки порядка $\varepsilon_{\text{machine}}$, что станет причиной изменения корней на значение порядка $\sqrt{\varepsilon_{\text{machine}}}$. К примеру, если $\varepsilon_{\text{machine}}$, корни вычисленного характеристического многочлена могут отклониться от истинных значений примерно, на 10^{-8}, что означает потерю восьми знаков в точности.

До того как читатель вычислит это собственными силами, необходимо учесть еще одну маленькую тонкость. Если используется алгоритм, специально разработанный для вычисления собственных значений единичной матрицы 2-го порядка, то возможно, что и вовсе не обнаружится

ошибок в вычислении, поскольку коэффициенты и корни рассматриваемого многочлена $\lambda^2 - 2\lambda + 1$ маленькие целые числа, которые будут точно представлены в компьютере. Однако, если провести эксперимент на слегка возмущенной матрице, такой как

$$A = \begin{pmatrix} 1 + 10^{-14} & 0 \\ 0 & 1 \end{pmatrix},$$

вычисленные собственные значения будут отличаться от своих истинных значений на величину ожидаемого порядка $\sqrt{\varepsilon_{\text{machine}}}$. Попробуйте!

5.4. Точность обратно устойчивого алгоритма

Предположим, имеется алгоритм $\tilde{g}$, обладающий обратной устойчивостью для задачи $g : X \to Y$. Насколько результаты, которые он обеспечивает, точны? Ответ зависит от числа обусловленности $\kappa = \kappa(x)$ задачи g. Если $\kappa(x)$ мало, то результат будет точен в относительном смысле, но если оно велико, точность будет падать пропорционально.

Теорема 5.1. *Пусть алгоритм $\tilde{g}$, обладающий обратной устойчивостью, применяется для решения задачи $g : X \to Y$ с числом обусловленности κ на компьютере, удовлетворяющим аксиомам 3.1 и 3.2. Тогда относительная ошибка удовлетворяет*

$$\frac{\|\tilde{g}(x) - g(x)\|}{\|g(x)\|} = O(\kappa(x)\varepsilon_{\text{machine}}). \tag{5.1}$$

Доказательство. По определению обратной устойчивости (4.5) имеем $\tilde{g}(x) = g(\tilde{x})$ для всех $\tilde{x} \in X$, удовлетворяющих

$$\frac{\|\tilde{x} - x\|}{\|x\|} = O(\varepsilon_{\text{machine}}).$$

По определению числа обусловленности

$$\kappa = \kappa(x) = \sup_{\delta x} \left(\frac{\|\delta g\|}{\|g(x)\|} \Big/ \frac{\|\delta x\|}{\|x\|} \right)$$

(здесь δx и δg – бесконечно малые величины) это означает, что выполняется

$$\frac{\|\tilde{g}(x) - g(x)\|}{\|g(x)\|} \le (\kappa(x) + o(1)) \frac{\|\tilde{x} - x\|}{\|x\|},$$

где $o(1)$ означает число, стремящееся к нулю, при $\varepsilon_{\text{machine}} \to 0$. Комбинируя эти выражения, получаем (5.1). □

5.5. Обратный анализ ошибок

Процесс, который был произведен в доказательстве теоремы 5.1 – это так называемый обратный анализ ошибок. Полученная точность соответствует двум шагам. Первый шаг – изучение обусловленности задачи, второй – изучение устойчивости компьютерного алгоритма. Окончательное заключение состоит в том, что если компьютерный алгоритм устойчив, тогда его конечная точность зависит от числа обусловленности задачи.

Чисто математически, это достаточно очевидно, но это не первая мысль, которая придет в голову неподготовленному человеку, если ему будет необходимо проанализировать численный алгоритм. Первая идея, которая придет ему в голову – это *прямой анализ ошибок*, т.е. анализ, где на каждом шаге алгоритма оцениваются ошибки вычисления и, таким образом, окончательная общая ошибка суммируется от шага к шагу.

Опыт показывает, что для большинства алгоритмов вычислительной линейной алгебры прямой анализ ошибок выполнить сложнее, чем обратный. Взглянув на уже изученное, несложно объяснить, почему это так. Пусть используется хорошо зарекомендовавший себя на практике алгоритм, скажем, для решения $Au = f$ на компьютере. Установленный факт, что полученные решения будут соответственно менее точными, если A – плохо обусловлена. Но как же прямой анализ ошибок может объяснить этот феномен? Число обусловленности A – настолько глобальное свойство, что оно более-менее невидимо на уровне индивидуальной ошибки операций с плавающей точкой, производимых для решения задачи $Au = f$. Тем не менее, так или иначе прямой анализ должен обнаружить это число обусловленности, для того чтобы прийти к корректному результату.

Короче, подтвержденный факт, что самые лучшие алгоритмы для большинства задач в общем случае дают результаты, близкие к точным, но уже для задач с небольшим возмущением исходных данных (относительно исходных). Метод обратного анализа ошибок позволяет достаточно аккуратно (с математической точки зрения) формализовать этот результат.

5.6. Упражнения

5.1. Для каждой из следующих задач предполагается, что описанные компьютерные алгоритмы удовлетворяют аксиомам 3.1 и 3.2. Для каждого из приведенных компьютерных алгоритмов указать, являются ли эти алгоритмы *обратно устойчивыми*, *устойчивыми, но не обратно устойчивыми* или *неустойчивыми* и доказать это или привести примеры, аргументирующие эти утверждения.
(a) Дано: $x \in \mathbb{C}$. Решение: $2x$, вычисляется как $x \oplus x$.
(b) Дано: $x \in \mathbb{C}$. Решение: x^2, вычисляется как $x \otimes x$.
(c) Дано: $x \in \mathbb{C} \setminus \{0\}$. Решение: 1, вычисляется как $x \oslash x$. (Компьютер удовлетворяющий (3.5), будет давать точный ответ, но предполагается, что компьютерная операция с плавающей точкой удовлетворяет (3.6)).
(d) Дано: $x \in \mathbb{C}$. Решение: 0, вычисляется как $x \ominus x$. (Опять, реальный компьютер может выполнять арифметические операции с плавающей точкой точнее, чем определено в (3.6)).
(e) Дано: нет. Решение: e, вычисляется суммированием $\sum_{k=0}^{\infty} 1/k!$ слева направо с использованием операций $\otimes$ и $\oplus$, останов производится когда изменение суммы $< \varepsilon_{\text{machine}}$.
(f) Дано: нет. Решение: e, вычисляется как в (e), но суммирование справа налево.

5.2. Докажите, что умножение двух матриц A и B с плавающей точкой, основанное на использовании скалярных произведений, удовлетворяет

$$\text{fl}(AB) = AB + E, \qquad |E| \le n\varepsilon_{\text{machine}}|A||B| + O(\varepsilon^2_{\text{machine}}).$$

5.3. Покажите, что если $F \in \mathbb{R}^{m\times n}$, где $m \ge n$, то $\| \, |F| \, \|_2 \le \sqrt{n}\|F\|_2$. Этот результат полезен при выводе оценок в терминах абсолютных величин элементов.

5.4. Пусть имеется функция извлечения квадратного корня, удовлетворяющая условию $\text{fl}(\sqrt{x}) = \sqrt{x}(1+\eta))$, где $|\eta| \le \varepsilon_{\text{machine}}$. Приведите алгоритмы для вычисления $\|x\|_2$ и оцените ошибки округления.

5.5. Пусть A и B – верхние треугольные матрицы порядка n из чисел с плавающей точкой. Если $\hat{C} = \text{fl}(AB)$ вычислено по любому из традиционных алгоритмов, верно ли, что $\hat{C} = \hat{A}\hat{B}$, где $\hat{A}$ и $\hat{B}$ близки к A и B?

5.6. Пусть A и B – $n \times n$-матрицы из чисел с плавающей точкой и матрица A невырожденная, причем $\| \, |A^{-1}||A| \, \|_\infty = \nu$. Покажите, что если $\hat{C} = \text{fl}(AB)$ вычислено по любому из традиционных алгоритмов,

то существует такая $\hat{B}$, что $\hat{C} = A\hat{B}$ и $\|\hat{B} - B\|_\infty \leq n\varepsilon_{\text{machine}}\nu\|B\|_\infty + O(\varepsilon^2_{\text{machine}})$.

5.7. Показать, используя результат упражнения 5.2, что

$$\|\text{fl}(AB) - AB\|_1 \leq n\varepsilon_{\text{machine}}\|A\|_1\|B\|_1 + O(\varepsilon^2_{\text{machine}}).$$

Библиографический список

[1] *Алберт А.* Регрессия, псевдоинверсия и рекуррентное оценивание: Пер. с англ. – М.: Наука, 1977. – 224 с.

[2] *Амосов А.А., Дубинский Ю.А., Копченова Н.В.* Вычислительные методы для инженеров. – М.: Высш. шк., 1994. – 544 с.

[3] *Беклемишев Д.В.* Дополнительные главы линейной алгебры: Учеб. пособие. – М.: Наука, 1983. – 336 с.

[4] *Беллман Р.* Введение в теорию матриц: Пер. с англ. – М.: Наука, 1976. – 368 с.

[5] *Воеводин В.В.* Вычислительные основы линейной алгебры. – М.: Наука, 1977.

[6] *Гантмахер Ф. Р.* Теория матриц. – М.: Наука, 1966. – 576 с.

[7] *Деммель Дж.* Вычислительная линейная алгебра. Теория и приложения: Пер. с англ. – М.: Мир, 2001. – 430 с.

[8] *Жданов А. И.* Прямой последовательный метод решения систем линейных алгебраических уравнений//Докл. РАН.–1997.–Т. 356.–№ 4.–С. 442–444.

[9] *Жданов А. И.* Регуляризация неустойчивых конечномерных линейных задач на основе расширенных систем//Ж. вычисл. матем. и матем. физ.–2005.–Т. 45.–№ 11.–С. 1918–1926.

[10] *Ильин В.А., Ким Г.Д.* Линейная алгебра и аналитическая геометрия: Учебник. – 2-е изд. – М.: Изд-во МГУ, 2002. – 320 с.

[11] *Ильин В.А., Садовничий В.А., Сендов Бл.Х.* Математический анализ: Учебник. – М.: Наука, 1979. – 720 с.

[12] *Малышев А.Н.* Введение в вычислительную линейную алгебру. – Новосибирск: Наука. Сиб. отд-ние. 1991. – 229 с.

[13] *Райс Дж.* Матричные вычисления и математическое обеспечение. – М.: Мир, 1984. – 264 с.

[14] *Рао С.Р.* Линейные статистические методы и их применения: Пер. с англ. – М.: Наука, 1968. – 548 с.

[15] *Самарский А.А., Гулин А.В.* Численные методы математической физики. – М.: Научный мир, 2000. – 316 с.

[16] *Тихонов А.Н., Арсенин В.Я.* Методы решения некорректных задач: Учебное пособие. – М.: Наука, 1986. – 288 с.

[17] *Тихонов А. Е., Гончарский А. В., Степанов В. В., Ягола А. Г.* Регуляризующие алгоритмы и априорная информация. – М.: Наука, 1983. – 200 с.

[18] *Шевцов Г.С.* Линейная алгебра: Учебное пособие. – М.: Гардарики, 1999. – 360 с.

[19] *Hensen P. Ch.* Regularization, GSVD and Truncated GSYD // BIT. 1989. V. 29. No. 3. P. 491–504.

[20] *Хорн Р., Джонсон Ч.* Матричный анализ: Пер. с англ. – М.: Мир, 1989. – 655 с.

[21] *Penrose R.* A generalized inverse for matrices // Proc. Cambridge Philos. Soc. 1955. V. 51. P. 406–413.

Жданов Александр Иванович

ВВЕДЕНИЕ В ВЫЧИСЛИТЕЛЬНУЮ ЛИНЕЙНУЮ АЛГЕБРУ

Самарский государственный
аэрокосмический университет.
443086, Самара, Московское шоссе, 34

Электронное учебное пособие

Самара 2011