



Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет имени
Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Робототехники и комплексной автоматизации»
КАФЕДРА «Системы автоматизированного проектирования (РК-6)»

ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №4 по дисциплине «Вычислительная математика»

Студент:	Рогалева Юлия Александровна
Группа:	РК6-52Б
Тип задания:	лабораторная работа
Тема:	Устойчивость прямых методов решения СЛАУ

Студент

подпись, дата

Рогалева Ю.А.

Фамилия, И.О.

Преподаватель

подпись, дата

Фамилия, И.О.

Москва, 2023

Содержание

Устойчивость прямых методов решения СЛАУ	3
Цель выполнения лабораторной работы	4
1 Реализация методов Гаусса для решения СЛАУ	4
2 Реализация метода прогонки для решения СЛАУ	6
3 Определение универсального метода	8
4 Разработка алгоритма генерации случайных невырожденных матриц . .	9
5 Анализ вычислительной устойчивости прямых методов решения СЛАУ на основе экспериментов со случайными матрицами	10
1 Разработка алгоритма генерации для получения положительно-определённых матриц	13
2 Реализация разложения Холецкого для решения СЛАУ	14
3 Сравнительный анализ метода Холецкого и универсального метода ре- шения СЛАУ	15
4 Анализ распределения спектральных радиусов и распределения чисел обусловленности	16
5 Влияние спектрального радиуса матрицы на вычислительную устойчи- вость алгоритмов	21
6 Оценка влияния соотношения максимального и минимального собствен- ных чисел на вычислительную устойчивость алгоритмов	23
7 Исследование влияния числа обусловленности на вычислительную устой- чивость решения СЛАУ	25
Заключение	27

Устойчивость прямых методов решения СЛАУ

Задание

Системы линейных алгебраических уравнений неизбежно появляются как промежуточный или конечный этап в нахождении численных решений в ряде методов вычислительной математики и анализа данных. В случае плотных матриц сравнительно небольшой размерности, для нахождения решения СЛАУ часто применяются прямые методы, такие как метод Гаусса, метод прогонки или разложение Холецкого. В то же время известно, что многие прямые методы обладают вычислительной неустойчивостью и могут приводить к некорректному решению для некоторых матриц коэффициентов. В этой лабораторной работе мы рассмотрим несколько видов матриц и с помощью генерации большого количества случайных матриц продемонстрируем наличие или отсутствие вычислительной неустойчивости у метода Гаусса, метода прогонки и разложения Холецкого.

Требуется (базовая часть):

1. Написать функцию `gauss(A, b, pivoting)`, которая возвращает решение СЛАУ $Ax = b$, полученное с помощью метода Гаусса. Если `pivoting=True`, то решение должно находиться с использованием выбора ведущего элемента. Если `pivoting=False`, то выбора ведущего элемента происходить не должно.
2. Написать функцию `thomas(A, b)`, которая возвращает решение СЛАУ $Ax = b$, полученное с помощью метода прогонки.
3. Среди реализованных методов, включая два варианта метода Гаусса, выбрать тот, который минимизирует вычислительные потери для случая квадратных матриц общего вида. В рамках задания такой метод будем называть “универсальным”.
4. Разработать и описать алгоритм генерации случайных невырожденных матриц размерности 6×6 с элементами $a_{ij} \in \mathbb{R}, |a_{ij}| < 1$ общего и 3-х диагонального вида.
5. Сгенерировав 1000 случайных матриц $A^{(j)}$ каждого типа с 32-битным float-представлением элементов необходимо провести следующий эксперимент:
 - (a) Выбрать “специальный” вычислительно-эффективный метод по типу матрицы.
 - (b) Для каждой СЛАУ $A^{(j)}x = [1, 1, 1, 1]^T$ найти решение с помощью “универсального” и “специального” методов, а затем найти относительную погрешность вычислений с помощью среднеквадратичной и супремум-нормы. Вывести на экран распределения погрешностей в виде гистограмм.
 - (c) Является ли выбранный “метод по умолчанию”, применённый к матрицам определённого класса, вычислительно устойчивым? Почему?

Требуется (продвинутая часть):

1. Расширить генератор из задания 4 для получения положительно-определенных матриц.
2. Написать функцию `cholesky(A, b)`, которая возвращает решение СЛАУ $Ax = b$, полученное с помощью разложения Холецкого.
3. Провести требуемый в задании 5 анализ учитывая метод Холецкого (сравнить с “универсальным” методом).
4. Для всех рассмотренных ранее матриц вывести на экран распределение спектральных радиусов и распределение чисел обусловленности в виде гистограмм. Сделать вывод.
5. Влияет ли значение спектрального радиуса матрицы на вычислительную устойчивость рассмотренных алгоритмов? Если да, то как?
6. Влияет ли значение отношения максимального по модулю собственного числа к минимальному по модулю собственному числу на вычислительную устойчивость алгоритма? Если да, то как?
7. Влияет ли число обусловленности на вычислительную устойчивость алгоритма? Если да, то как?

Цель выполнения лабораторной работы

Цель выполнения лабораторной работы: исследование и сравнение вычислительной устойчивости различных прямых методов решения систем линейных алгебраических уравнений (СЛАУ), таких как метод Гаусса, метод прогонки и разложение Холецкого, на основе анализа ошибок при работе с матрицами разных типов.

1 Реализация методов Гаусса для решения СЛАУ

Метод Гаусса является базовым методом решения СЛАУ и состоит в приведении матрицы A к треугольному виду с помощью элементарных преобразований, после чего решение x находится сравнительно легко.

Этап прямого хода метода Гаусса:

Расширенная матрица имеет вид:

$$\tilde{A} = [A, b] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{array} \right]. \quad (1)$$

Для того, чтобы привести матрицу \mathbf{A} к треугольному виду, необходимо последовательно обнулять элементы, находящиеся под главной диагональю:

$$(i) \longrightarrow (i) - \frac{a_{i1}}{a_{11}}(1), \quad i = 2, \dots, n \quad (2)$$

где (i) обозначает i -ю строку. После применения этой операции к расширенной матрице получается:

$$\tilde{\mathbf{A}} = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{n2}^{(1)} & \dots & a_{nn}^{(1)} & b_n^{(1)} \end{array} \right], \quad (3)$$

где $a_{ij}^{(1)} = a_{ij} - \frac{a_{i1}}{a_{11}}a_{1j}$ и $b_i^{(1)} = b_i - \frac{a_{i1}}{a_{11}}b_1$. Аналогично, для обнуления элементов, находящихся под диагональным элементом $a_{22}^{(1)}$, необходимо применить следующее элементарное преобразование:

$$(i) \longrightarrow (i) - \frac{a_{i2}}{a_{22}^{(1)}}(2), \quad i = 3, \dots, n \quad (4)$$

Применяя подобные преобразования каскадно вплоть до последнего диагонального элемента, получается верхняя треугольная матрица для \mathbf{A} :

$$\tilde{\mathbf{A}} = \left[\begin{array}{cccc|c|c} a_{11} & a_{12} & \dots & \dots & a_{1n} & b_1 \\ 0 & a_{22}^{(1)} & \dots & \dots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & a_{n-1,n-1}^{(n-2)} & a_{n-1,n}^{(n-2)} & b_n^{(n-2)} \\ 0 & 0 & \dots & 0 & a_{nn}^{(n-1)} & b_n^{(n-1)} \end{array} \right]. \quad (5)$$

Этап обратного хода метода Гаусса:

Решение СЛАУ, в которой матрица имеет треугольную форму, можно найти с помощью обратного хода метода Гаусса:

$$\begin{aligned} x_n &= \frac{b_n^{(n-1)}}{a_{nn}^{(n-1)}}, \\ x_{n-1} &= \frac{b_{n-1}^{(n-2)} - a_{n-1,n}^{(n-2)}x_n}{a_{n-1,n-1}^{(n-2)}}, \\ &\dots \\ x_1 &= \frac{b_1 - \sum_{i=2}^n a_{1i}x_i}{a_{11}}. \end{aligned}$$

Метод Гаусса с выбором главного элемента

В случае, когда на k -й итерации алгоритма диагональный элемент оказывается равен нулю, рассчитать множитель $m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$ невозможно. Эта проблема решается соответствующей перестановкой строк. Чтобы избежать этого и уменьшить ошибки

округления при $a_{kk}^{(k)} \approx 0$, производится перестановка строк, выбирая в качестве $a_{kk}^{(k)}$ максимальный по модулю элемент из k -го столбца:

$$|a_{kk}^{(k)}| = \max_{k \leq i \leq n} |a_{ik}^{(k)}|. \quad (6)$$

Такой подход называется частичным выбором главного элемента.

Листинг 1. Функция, реализующая метод Гаусса с частичным выбором и без выбора главного элемента

```

1 def gauss(A, b, pivoting=False):
2     n = A.shape[0]
3     x = np.zeros(n, dtype=np.float32)
4     aug_matrix = np.hstack((A.astype(np.float32), b.astype(np.float32).reshape(-1, 1)))
5     for k in range(n):
6         if pivoting:
7             max_row = k + np.argmax(np.abs(aug_matrix[k:, k]))
8             aug_matrix[[k, max_row]] = aug_matrix[[max_row, k]]
9         for i in range(k + 1, n):
10            factor = aug_matrix[i, k] / aug_matrix[k, k]
11            aug_matrix[i, k:] -= factor * aug_matrix[k, k:]
12    for i in range(n - 1, -1, -1):
13        sum_val = np.dot(aug_matrix[i, i + 1:n], x[i + 1:n])
14        x[i] = (aug_matrix[i, -1] - sum_val) / aug_matrix[i, i]
15    return x.reshape(-1, 1)

```

2 Реализация метода прогонки для решения СЛАУ

Метод прогонки является специальным методом решения СЛАУ, имеющих трехдиагональную матрицу. Имеется следующее матричное уравнение:

$$\begin{bmatrix} a_{11} & a_{12} & 0 & \dots & \dots & \dots & 0 \\ a_{21} & a_{22} & a_{23} & \ddots & & & \vdots \\ 0 & a_{32} & a_{33} & a_{34} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & a_{n-1,n-2} & a_{n-1,n-1} & a_{n-1,n} \\ 0 & \dots & \dots & \dots & 0 & a_{n,n-1} & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}, \quad (7)$$

где каждая строка эквивалентна рекуррентному соотношению вида:

$$a_{i,i-1}x_{i-1} + a_{ii}x_i + a_{i,i+1}x_{i+1} = b_i. \quad (8)$$

Приведение СЛАУ (7) к верхней треугольной форме методом последовательного исключения:

$$\begin{bmatrix} \tilde{a}_{11} & \tilde{a}_{12} & 0 & \dots & \dots & 0 \\ 0 & \tilde{a}_{22} & \tilde{a}_{23} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \tilde{a}_{n-1,n-1} & \tilde{a}_{n-1,n} \\ 0 & \dots & \dots & \dots & 0 & \tilde{a}_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \vdots \\ \vdots \\ \tilde{b}_{n-1} \\ \tilde{b}_n \end{bmatrix}. \quad (9)$$

Решение для x_{i-1} , выраженное через x_i :

$$x_{i-1} = \frac{\tilde{b}_{i-1} - \tilde{a}_{i-1,i}x_i}{\tilde{a}_{i-1,i-1}}, \quad (10)$$

задает выражение для обратного хода метода Гаусса. Вместо явного вывода неизвестных коэффициентов, строится рекуррентное соотношение, позволяющее рекурсивно найти нужные коэффициенты. Пусть:

$$x_{i-1} = \gamma_i x_i + \beta_i. \quad (11)$$

Тогда подстановка в (8) дает:

$$\begin{aligned} & a_{i,i-1}(\gamma_i x_i + \beta_i) + a_{ii}x_i + a_{i,i+1}x_{i+1} = b_i \\ \implies & x_i = \frac{-a_{i,i+1}}{a_{i,i-1}\gamma_i + a_{ii}}x_{i+1} + \frac{b_i - a_{i,i-1}\beta_i}{a_{i,i-1}\gamma_i + a_{ii}}. \end{aligned} \quad (12)$$

Таким образом, из (11) и (12) получаются следующие рекуррентные соотношения для определения γ_i и β_i :

$$\gamma_{i+1} = \frac{-a_{i,i+1}}{a_{i,i-1}\gamma_i + a_{ii}}, \quad (13)$$

$$\beta_{i+1} = \frac{b_i - a_{i,i-1}\beta_i}{a_{i,i-1}\gamma_i + a_{ii}}. \quad (14)$$

Последовательно вычислив все коэффициенты γ_i и β_i , решение СЛАУ находится с помощью обратного хода метода Гаусса по формуле (11). Для завершения построения метода прогонки достаточно найти выражения для коэффициентов γ_i , β_i и неизвестной x_n . Уравнение для x_1 имеет вид:

$$a_{11}x_1 + a_{12}x_2 = b_1 \implies x_1 = -\frac{a_{12}}{a_{11}}x_2 + \frac{b_1}{a_{11}}. \quad (15)$$

Тогда сравнение с (12) дает:

$$\frac{-a_{12}}{a_{10}\gamma_1 + a_{11}} = -\frac{a_{12}}{a_{11}}, \quad (16)$$

$$\frac{b_1 - a_{10}\beta_1}{a_{10}\gamma_1 + a_{11}} = \frac{b_1}{a_{11}}, \quad (17)$$

из чего следует $\gamma_1 = \beta_1 = 0$ (или, эквивалентно, $a_{10} = 0$). Уравнение, соответствующее Последней строке матрицы, определяет x_n :

$$a_{n,n-1}x_{n-1} + a_{nn}x_n = b_n \implies x_n = -\frac{a_{n,n-1}}{a_{nn}}x_{n-1} + \frac{b_n}{a_{nn}}. \quad (18)$$

Подстановка в уравнение (11), записанное для $i = n$ дает:

$$x_{n-1} = \gamma_n x_n + \beta_n \implies x_n = \frac{b_n - a_{n,n-1}\beta_n}{a_{nn} + a_{n,n-1}\gamma_n}. \quad (19)$$

Листинг 2. Функция, реализующая метод прогонки

```

1 def thomas(A, b):
2     n = len(b)
3     gamma = np.zeros(n, dtype=np.float32)
4     beta = np.zeros(n, dtype=np.float32)
5     x = np.zeros(n, dtype=np.float32)
6     for i in range(n - 1):
7         denominator = A[i, i] + gamma[i] * A[i, i + 1]
8         gamma[i + 1] = -A[i, i + 1] / denominator
9         beta[i + 1] = (b[i] - beta[i] * A[i, i + 1]) / denominator
10    x[-1] = (b[-1] - beta[-1] * A[-1, -2]) / (A[-1, -1] + gamma[-1] * A[-1, -2])
11    for i in range(n - 2, -1, -1):
12        x[i] = gamma[i + 1] * x[i + 1] + beta[i + 1]
13    return x

```

3 Определение универсального метода

Для определения метода, минимизирующего вычислительные погрешности при решении систем линейных алгебраических уравнений (СЛАУ) вида $\mathbf{Ax} = \mathbf{b}$, где \mathbf{A} — квадратная матрица общего вида, и \mathbf{b} — вектор правых частей, выполнен следующий анализ:

1. Сгенерированы $n = 1000$ тестовых пар (\mathbf{A}, \mathbf{b}) , где $\mathbf{A} \in \mathbb{R}^{m \times m}$ — случайная квадратная матрица, а $\mathbf{b} \in \mathbb{R}^m$ — соответствующий вектор правых частей. Матрицы и векторы сгенерированы с использованием равномерного распределения вещественных чисел.
2. СЛАУ решены с помощью всех реализованных в предыдущих пунктах методов.
3. Определена погрешность решения как евклидова норма разности между исходным вектором правых частей и вектором, полученным подстановкой решения обратно в уравнение: $\|\Delta \mathbf{b}\| = \|\mathbf{Ax} - \mathbf{b}\|$.

4. Для каждого метода рассчитана средняя погрешность по всем тестовым примерам. Метод, показавший наименьшую среднюю погрешность, считается "точным" для данного исследования. Полученные следующие значения погрешностей:

- метод Гаусса без выбора главного элемента: 2.1733427,
- метод Гаусса с выбором главного элемента: 2.17335,
- метод прогонки: 11.411875.

На основе проведенных экспериментов, метод Гаусса с выбором главного элемента продемонстрировал **наименьшую среднюю погрешность**, что делает его предпочтительным для решения СЛАУ общего вида в контексте минимизации вычислительных погрешностей. В рамках лабораторной работы этот метод будет называться "универсальным". Так как эксперимент проводился на матрицах общего вида, то метод прогонки применялся и к нетрехдиагональным матрицам, что сильно увеличило погрешность.

4 Разработка алгоритма генерации случайных невырожденных матриц

Алгоритм генерации случайных невырожденных вещественнозначных матриц размерности 6×6 с элементами $|a_{ij}| < 1$ включает в себя выбор распределения для элементов a_{ij} , порядок генерации элементов, а также механизм обеспечения невырожденности матрицы. Функция `generate_matrix`, описанная в листинге 3 создаёт невырожденную матрицу заданного размера, элементы которой равномерно распределены в интервале $[-1, 1]$. При указании параметра `tridiagonal` генерируется трехдиагональная матрица с случайными элементами вдоль главной диагонали и соседних с ней. Для обеспечения невырожденности матрицы проверяется, что её определитель не равен нулю. В случае вырожденной матрицы процедура повторяется, пока не будет получена невырожденная матрица.

Листинг 3. Алгоритм генерации случайных невырожденных матриц общего и 3-х диагонального вида

```
1 def generate_matrix(size=6, tridiagonal=False):
2     while True:
3         if tridiagonal:
4             A = np.zeros((size, size), dtype=np.float32)
5             for i in range(size):
6                 A[i, i] = np.random.uniform(-1, 1)
7                 if i > 0:
8                     A[i, i - 1] = A[i - 1, i] = np.random.uniform(-1, 1)
9         else:
10            A = np.random.uniform(-1, 1, (size, size)).astype(np.float32)
11            if np.linalg.det(A) != 0:
12                return A
```

5 Анализ вычислительной устойчивости прямых методов решения СЛАУ на основе экспериментов со случайными матрицами

Используя алгоритм, представленный в листинге 3, сгенерировано 1000 случайных матриц $\mathbf{A}^{(j)}$.

- а) Вычисление количества операций для решения СЛАУ методом Гаусса фокусируется на умножениях и делениях, так как они требуют больше времени по сравнению со сложениями и вычитаниями.

На каждой k -й из $(n - 1)$ итераций прямого хода выполняется $(n - k)$ делений для нахождения множителя $m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$. Далее, для $(n - k)$ строк происходит $(n - k + 1)$ умножений, что в сумме дает $(n - k)(n - k + 1)$ операций на k -й итерации. Общее количество умножений и делений на этом этапе составляет $(n - k)(n - k + 2)$. Суммируя по всем k , получается общее количество умножений и делений в прямом ходе метода Гаусса:

$$\begin{aligned} \sum_{k=1}^{n-1} (n - k)(n - k + 2) &= \sum_{k=1}^{n-1} (n - k)^2 + 2 \sum_{k=1}^{n-1} (n - k) = \sum_{k=1}^{n-1} k^2 + 2 \sum_{k=1}^{n-1} k = \\ &= \frac{(n - 1)n(2n - 1)}{6} + 2 \frac{(n - 1)n}{2} = \frac{2n^3 + 3n^2 - 5n}{6}. \end{aligned} \quad (20)$$

В обратном ходе нахождение x_k требует $n - k$ умножений и одного деления, общее количество операций равно:

$$\sum_{k=1}^n (n - k + 1) = \sum_{k=1}^n k = \frac{n(n + 1)}{2} = \frac{n^2 + n}{2}. \quad (21)$$

Суммарное число операций для метода Гаусса:

$$\frac{2n^3 + 3n^2 - 5n}{6} + \frac{n^2 + n}{2} = O(n^3). \quad (22)$$

Метод Гаусса с выбором главного элемента, требующий $O(n^3)$ сравнений, подходит для задач с высокой чувствительностью к ошибкам округления.

Используя рекуррентные соотношения (11), (14) и (13), можно найти решение СЛАУ с помощью $O(n)$ операций. Более того, если исходная матрица обладает свойством строгого диагонального преобладания, то метод прогонки является вычислительно устойчивым.

Таким образом, "специальным" методом будет называться метод прогонки.

- б) Для каждой СЛАУ $\mathbf{A}^{(j)} \mathbf{x} = [1, 1, 1, 1, 1, 1]^T$ найдено решение с помощью "универсального" и "специального" методов, представленных в листингах 1 и 2.

Определение 1. Относительная погрешность

Относительной погрешностью приближенного значения a^* называют величину $\delta(a^*)$, которая определена как:

$$\delta(a^*) = \left| \frac{a - a^*}{a} \right|, \quad (23)$$

где a — точное значение.

Вычисление среднеквадратичной нормы производится по формуле (24).

$$\|\mathbf{x}\|_2 = \left(\sum_{i=1}^n x_i^2 \right)^{\frac{1}{2}}. \quad (24)$$

Распределение относительных погрешностей с помощью среднеквадратичной нормы (24) для матриц общего вида в виде гистограмм представлено на рисунке 1.

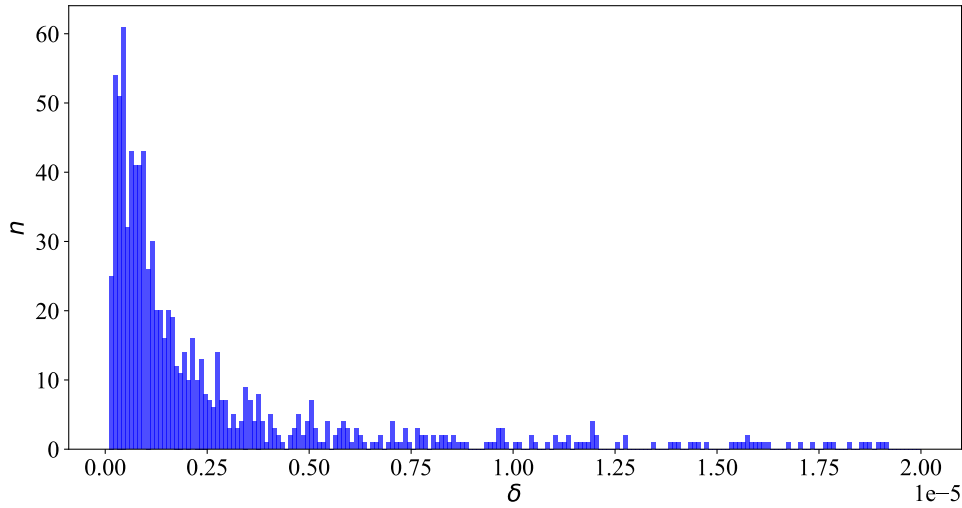


Рис. 1. График распределения погрешностей вычислений с помощью среднеквадратичной нормы для матриц общего вида

Из графика, представленного на рисунке 1, видно, что распределение плотно и сосредоточено вокруг низкого значения погрешности. Это указывает на высокую точность и надёжность вычислительного метода.

Для нахождения супремум-нормы нужно воспользоваться формулой (25).

$$\|\mathbf{x}\|_\infty = \max_{i \in [1;n]} |x_i|. \quad (25)$$

Распределение погрешностей с помощью супремум-нормы для матриц общего вида в виде гистограмм представлено на рисунке 2.

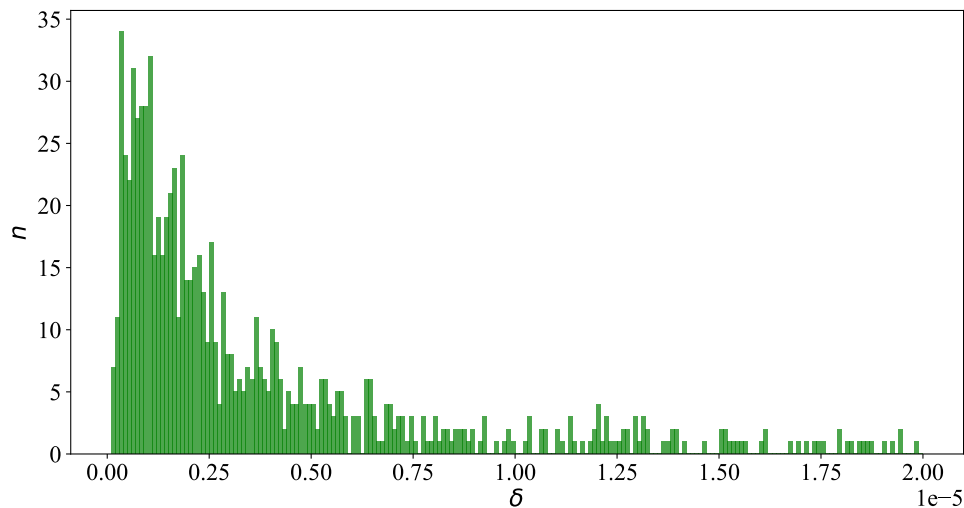


Рис. 2. График распределения погрешностей вычислений с помощью супремум-нормы для матриц общего вида

Из графика, представленного на рисунке 2 видно, что распределение относительной погрешности δ концентрируется около низких значений. Это говорит о том, что даже в наихудших сценариях вычислительный метод остаётся достаточно точным.

с) Метод прогонки является вычислительно устойчивым для решения СЛАУ, когда используемая матрица обладает определенными свойствами. Этот метод особенно эффективен для трехдиагональных матриц, которые являются строго диагонально доминирующими или симметричными и положительно определенными (рисунки 3 и 4).

- Если матрица строго диагонально доминирующая, то есть каждый элемент на главной диагонали по абсолютной величине больше суммы абсолютных величин всех других элементов в его строке, метод прогонки гарантированно устойчив. Это обусловлено тем, что большие диагональные элементы уменьшают влияние ошибок округления при выполнении операций.
- Для симметричных и положительно определенных трехдиагональных матриц метод прогонки также устойчив, так как он не усиливает ошибки округления в значительной степени, и каждый шаг метода сохраняет положительное определение подматриц.

Однако для матриц, не удовлетворяющих этим условиям, метод прогонки может быть неустойчивым из-за возможного накопления ошибок округления, особенно в случаях малых или нулевых диагональных элементов. В таких случаях требуется дополнительный анализ или использование других методов решения.

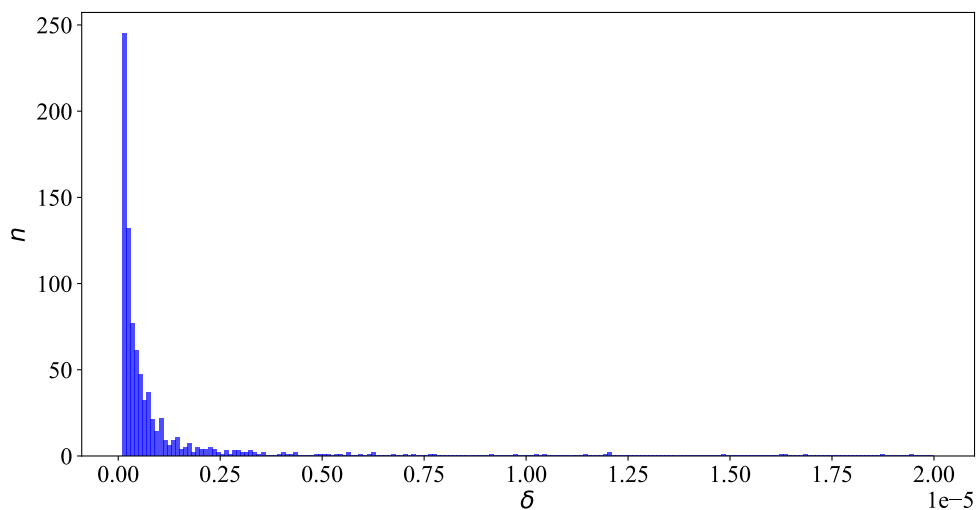


Рис. 3. График распределения погрешностей вычислений с помощью среднеквадратичной нормы для матриц 3-х диагонального вида

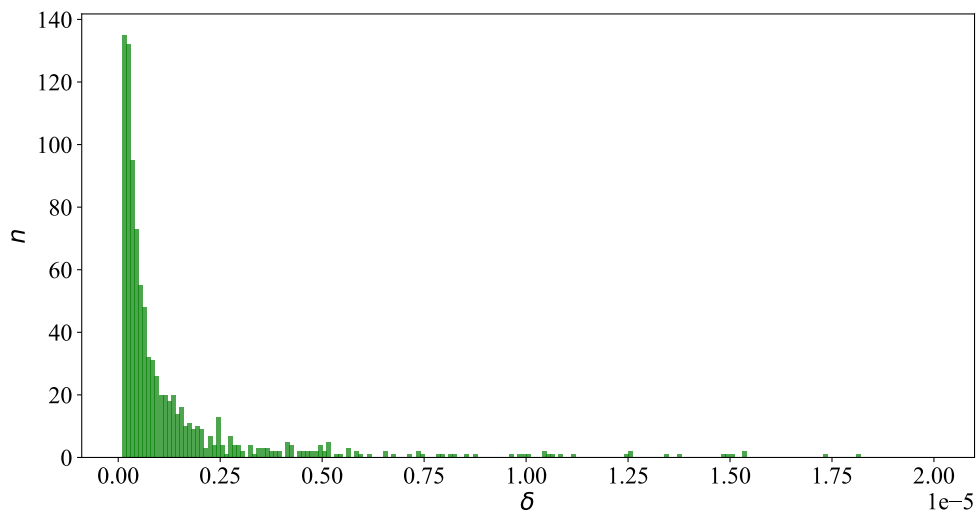


Рис. 4. График распределения погрешностей вычислений с помощью супремум-нормы для матриц 3-х диагонального вида

1 Разработка алгоритма генерации для получения положительно-определённых матриц

Алгоритм начинается с генерации случайной симметричной матрицы, элементы которой находятся в диапазоне от -1 до 1. Затем он добавляет к этой матрице кратное

единичной матрицы, чтобы гарантировать положительную определённость. Алгоритм проверяет полученную матрицу на положительную определённость через собственные значения и на достаточный модуль определителя. Процесс повторяется, пока не будет найдена подходящая матрица. Код алгоритма продемонстрирован в листинге 4.

Листинг 4. Алгоритм генерации случайных положительно-определённых матриц

```

1 def generate_symmetric_matrix(size):
2     random_matrix = np.random.rand(size, size).astype(np.float32) * 2 - 1
3     return (random_matrix + random_matrix.T) / 2
4 def is_positive_definite(matrix):
5     return np.all(np.linalg.eigvals(matrix) > 0)
6 def generate_positive_definite_matrix(dimension):
7     while True:
8         symmetric_matrix = generate_symmetric_matrix(dimension)
9         pd_matrix = symmetric_matrix + np.eye(dimension) * dimension
10        if is_positive_definite(pd_matrix) and np.abs(np.linalg.det(pd_matrix)) > 0.01:
11            return pd_matrix

```

2 Реализация разложения Холецкого для решения СЛАУ

Теорема 1. Разложение Холецкого

Матрица A является положительно определенной тогда и только тогда, когда существует разложение $A = LL^T$, называемое разложением Холецкого, где L — нижняя треугольная матрица с ненулевыми элементами на диагонали.

Реализация данного метода представлена в листинге 5.

Листинг 5. Функции, реализующие метод Холецкого

```

1 def cholesky_decomposition(A):
2     n = A.shape[0]
3     L = np.zeros_like(A, dtype=np.float32)
4     for i in range(n):
5         for j in range(i+1):
6             temp_sum = np.dot(L[i, :j], L[j, :j])
7             if i == j:
8                 L[i, j] = np.sqrt(A[i, i] - temp_sum)
9             else:
10                L[i, j] = (A[i, j] - temp_sum) / L[j, j]
11    return L
12
13 def cholesky(A, b):
14     L = cholesky_decomposition(A)
15     y = np.linalg.solve(L, b)
16     x = np.linalg.solve(L.T, y)
17    return x

```

3 Сравнительный анализ метода Холецкого и универсального метода решения СЛАУ

Для каждой СЛАУ $A^{(j)}x = [1, 1, 1, 1, 1, 1]^T$ найдено решение с помощью "универсального" метода и разложения Холецкого, представленных в листингах 1 и 5 соответственно.

Из графиков, демонстрирующих распределения относительной погрешности (23) с помощью среднеквадратичной нормы (24) (рисунок 5) и с помощью супремум-нормы (25) (рисунок 6), видно, что она сосредоточена в узком диапазоне. Это говорит о стабильности метода для матриц данного вида.

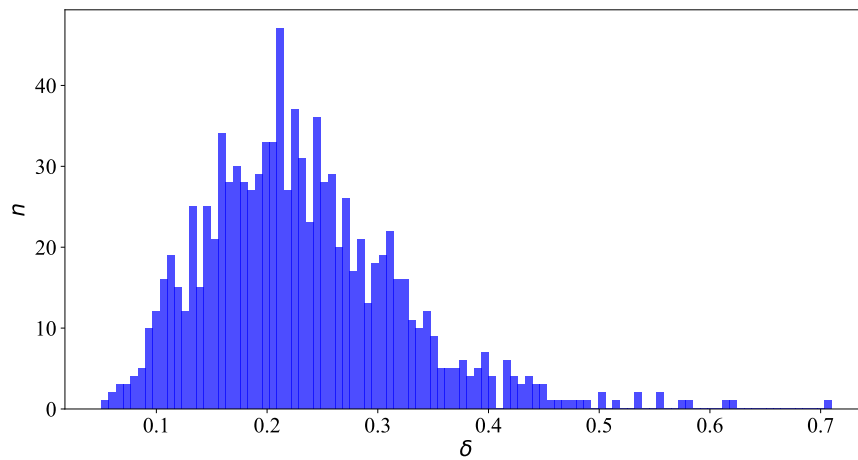


Рис. 5. График распределения погрешностей вычислений с помощью среднеквадратичной нормы для положительно-определённых матриц

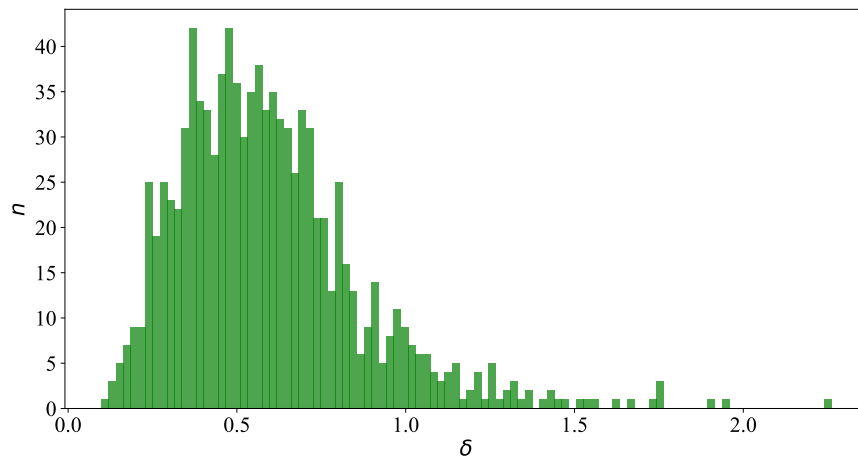


Рис. 6. График распределения погрешностей вычислений с помощью супремум-нормы для положительно-определённых матриц

4 Анализ распределения спектральных радиусов и распределения чисел обусловленности

Определение 2. Спектральный радиус

Спектральным радиусом матрицы \mathbf{A} называется число $\rho(\mathbf{A}) \in \mathbb{R}$ такое, что:

$$\rho(\mathbf{A}) = \max_{i \in [1, m]} |\lambda_i|, \quad (26)$$

где λ_i — одно из m собственных чисел матрицы \mathbf{A} .

Распределение спектральных радиусов (26) для матриц общего вида в виде гистограмм представлено на рисунке 7.

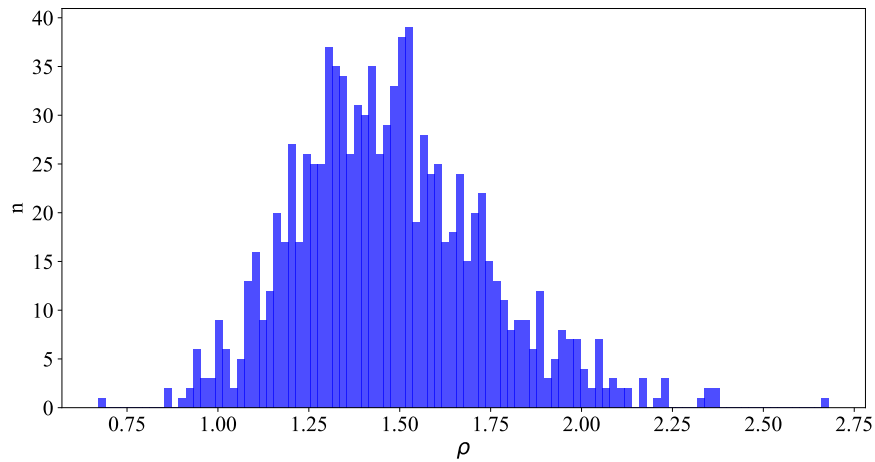


Рис. 7. График распределения спектральных радиусов для матриц общего вида

Распределение спектральных радиусов (26) для матриц 3-х диагонального вида в виде гистограмм представлено на рисунке 8.

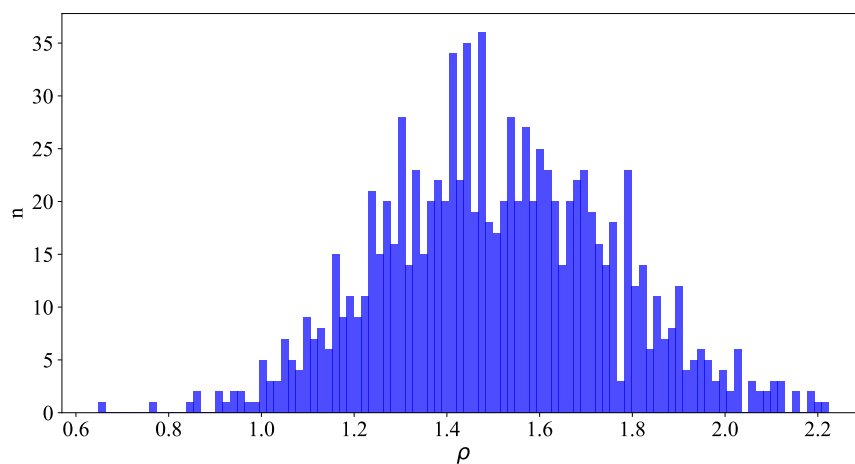


Рис. 8. График распределения спектральных радиусов для матриц 3-х диагонального вида

Распределение спектральных радиусов (26) для положительно-определённых матриц в виде гистограмм представлено на рисунке 9.

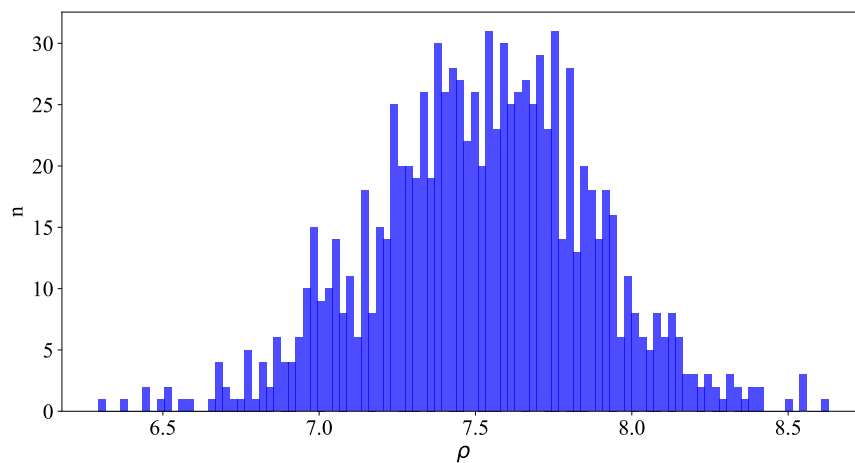


Рис. 9. График распределения спектральных радиусов для положительно-определённых матриц

Анализируя гистограммы спектральных радиусов для трёх разных типов матриц — общего вида, тридиагональных и положительно-определённых — можно сделать следующие выводы:

1. Матрицы общего вида:

- спектральные радиусы этих матриц имеют более широкий диапазон значений, так как нет структурных ограничений на распределение элементов

матрицы,

- распределение спектральных радиусов более разнообразное и менее предсказуемое, что отражает большую свободу в выборе элементов матрицы.

2. Трёхдиагональные матрицы:

- трёхдиагональные матрицы имеют более ограниченную структуру, что может привести к более узкому и предсказуемому распределению спектральных радиусов,
- спектральные радиусы этих матриц менее разнообразные по сравнению с общими матрицами, отражая структурные ограничения трёхдиагональности.

3. Положительно-определённые матрицы:

- положительно-определённые матрицы характеризуются положительными собственными значениями, что приводит к спектральным радиусам, сосредоточенным в положительной области,
- распределение спектральных радиусов более сконцентрированное и сдвинутое в сторону больших значений из-за добавления больших чисел на диагональ для обеспечения положительной определенности.

Распределение спектральных радиусов значительно зависит от структуры и свойств матриц. Матрицы общего вида представляют наибольшее разнообразие в спектральных радиусах из-за меньших структурных ограничений. Трёхдиагональные матрицы показывают более предсказуемое распределение из-за своих структурных ограничений. Положительно-определённые матрицы имеют тенденцию к большим значениям спектральных радиусов, что обусловлено их положительной определенностью и структурой.

Определение 3. Вектор невязки

Пусть $\tilde{\mathbf{x}} \in \mathbb{R}^n$ является приближением к точному решению СЛАУ $\mathbf{Ax} = \mathbf{b}$. Тогда вектором невязки называется $\mathbf{r} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}$.

Теорема 2. Число обусловленности

Пусть $\tilde{\mathbf{x}} \in \mathbb{R}^n$ является приближением к точному решению СЛАУ $\mathbf{Ax} = \mathbf{b}$, \mathbf{A} — невырожденная матрица и \mathbf{r} — вектор невязки. Тогда для любой индуцированной матричной нормы верно:

$$\|\mathbf{x} - \tilde{\mathbf{x}}\| \leq \|\mathbf{r}\| \cdot \|\mathbf{A}^{-1}\|, \quad (27)$$

и при $\mathbf{x}, \mathbf{b} \neq \mathbf{0}$:

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq K(\mathbf{A}) \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}, \quad (28)$$

где $K(\mathbf{A}) = \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\|$ называется числом обусловленности.

Распределение чисел обусловленности (28) для матриц общего вида в виде гистограмм представлено на рисунке 10.

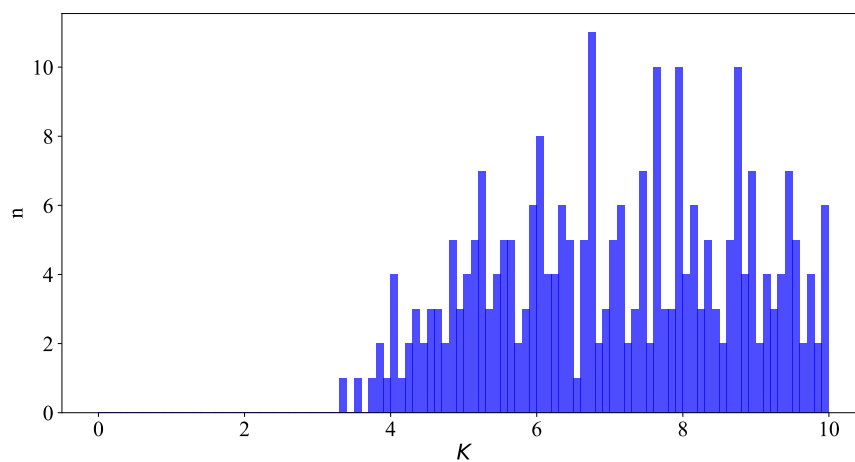


Рис. 10. График распределения чисел обусловленности для матриц общего вида

Распределение чисел обусловленности (28) для матриц 3-х диагонального вида в виде гистограмм представлено на рисунке 11.

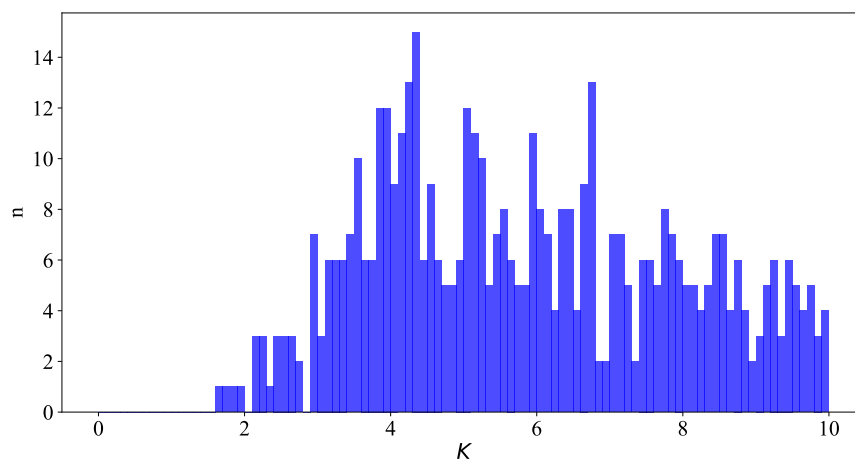


Рис. 11. График распределения чисел обусловленности для матриц 3-х диагонального вида

Распределение чисел обусловленности (28) для положительно-определённых матриц в виде гистограмм представлено на рисунке 12.

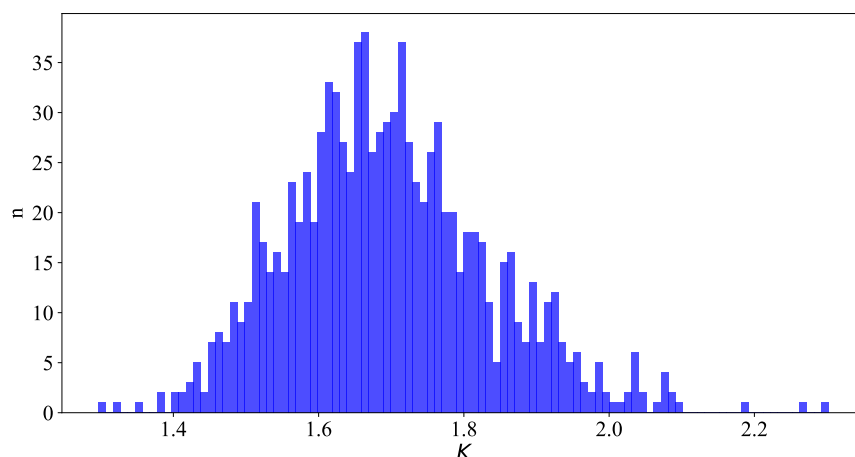


Рис. 12. График распределения чисел обусловленности для положительно-определённых матриц

Анализируя гистограммы чисел обусловленности для трёх разных типов матриц — общего вида, тридиагональных и положительно-определённых — можно сделать следующие выводы:

1. Матрицы общего вида:

- числа обусловленности для этих матриц могут показывать большое разнообразие, отражая широкий спектр возможных структур и характеристик матриц,
- можно ожидать более широкого диапазона значений чисел обусловленности, включая как очень малые, так и очень большие значения, что указывает на смешение хорошо и плохо обусловленных матриц.

2. Трёхдиагональные матрицы:

- распределение чисел обусловленности для тридиагональных матриц, возможно, более узкое и предсказуемое из-за структурных ограничений, налагаемых их тридиагональной формой,
- эти числа могут быть более стабильными и с меньшей вероятностью достигать крайне высоких значений, что делает тридиагональные матрицы потенциально более устойчивыми для вычислений.

3. Положительно-определённые матрицы:

- числа обусловленности для положительно-определённых матриц могут быть относительно низкими и более сгруппированными, отражая их хорошие численные свойства,

- положительная определённость матрицы обычно сопровождается более стабильными численными характеристиками, что снижает риск чувствительности к небольшим изменениям в данных.

Числа обусловленности варьируются в зависимости от структуры и свойств матриц. Матрицы общего вида показывают наибольшую вариабельность в числах обусловленности. Тридиагональные матрицы, благодаря своей структуре, часто имеют более умеренные и предсказуемые числа обусловленности. Положительно-определённые матрицы, как правило, обладают лучшими численными свойствами, что отражается в их более низких и сгруппированных числах обусловленности.

5 Влияние спектрального радиуса матрицы на вычислительную устойчивость алгоритмов

Для анализа влияния спектрального радиуса матрицы на вычислительную устойчивость алгоритмов построены графики $\delta(\rho)$ для всех видов рассмотренных матриц (рисунки 13-15).

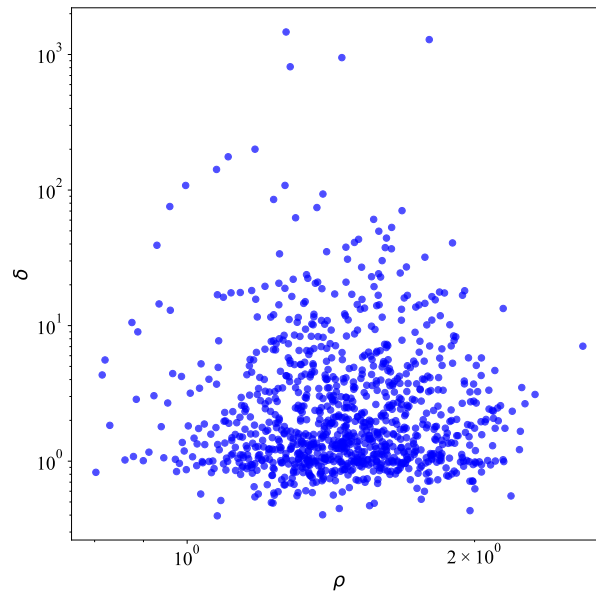


Рис. 13. Влияние спектрального радиуса матрицы на вычислительную устойчивость алгоритмов для матриц общего вида

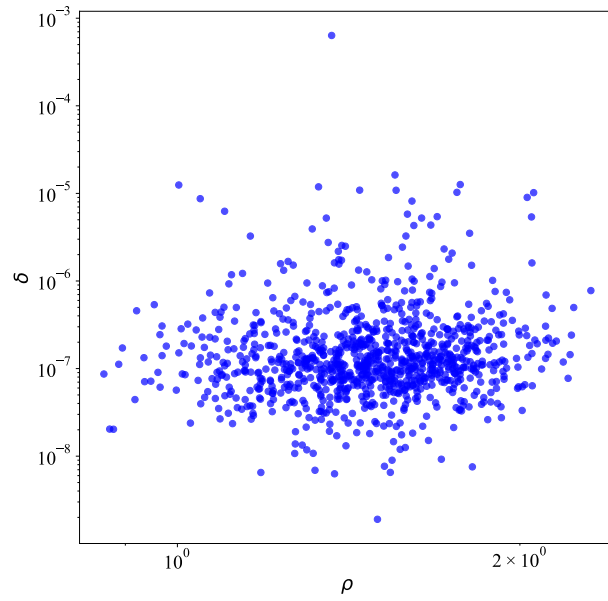


Рис. 14. Влияние спектрального радиуса матрицы на вычислительную устойчивость алгоритмов для матриц 3-х диагонального вида

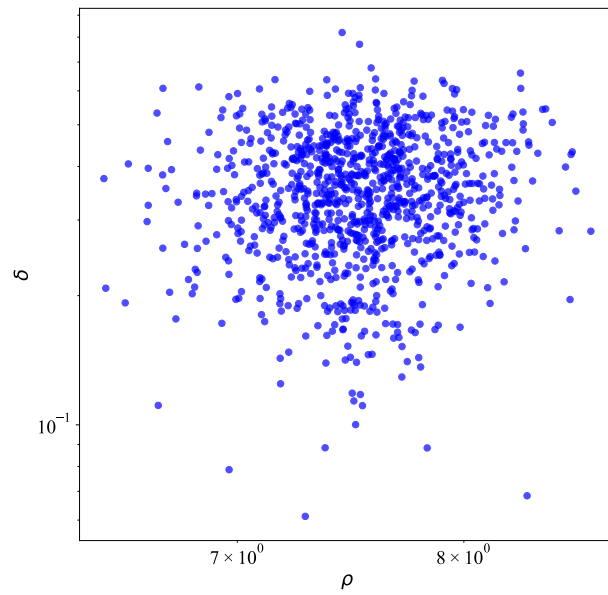


Рис. 15. Влияние спектрального радиуса матрицы на вычислительную устойчивость алгоритмов для положительно-определённых матриц

Хотя спектральный радиус $\rho(\mathbf{A})$ не является прямым индикатором устойчивости для методов Гаусса и Холецкого, он может выступать в роли индикатора потенциальных проблем, связанных с числом обусловленности матрицы $K(\mathbf{A})$. Для метода Гаусса, особенно без выбора главного элемента, высокий $\rho(\mathbf{A})$ может указывать на повышенный риск численной неустойчивости. При использовании метода Гаусса с выбором главного элемента увеличивается устойчивость к численным ошибкам. В контексте метода Холецкого, высокий $\rho(\mathbf{A})$ для положительно-определённых матриц может быть более значимым, указывая на повышенную чувствительность к ошибкам входных данных.

6 Оценка влияния соотношения максимального и минимального собственных чисел на вычислительную устойчивость алгоритмов

Для оценки влияния соотношения максимального и минимального собственных чисел на вычислительную устойчивость алгоритмов построены графики $\delta(\frac{|\lambda_{max}|}{|\lambda_{min}|})$ для всех видов рассмотренных матриц (рисунки 16-18).

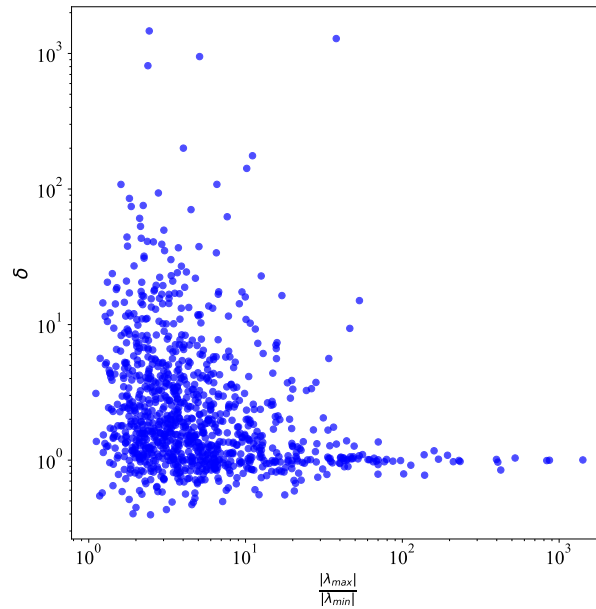


Рис. 16. Влияние соотношения максимального и минимального собственных чисел матрицы на вычислительную устойчивость алгоритмов для матриц общего вида

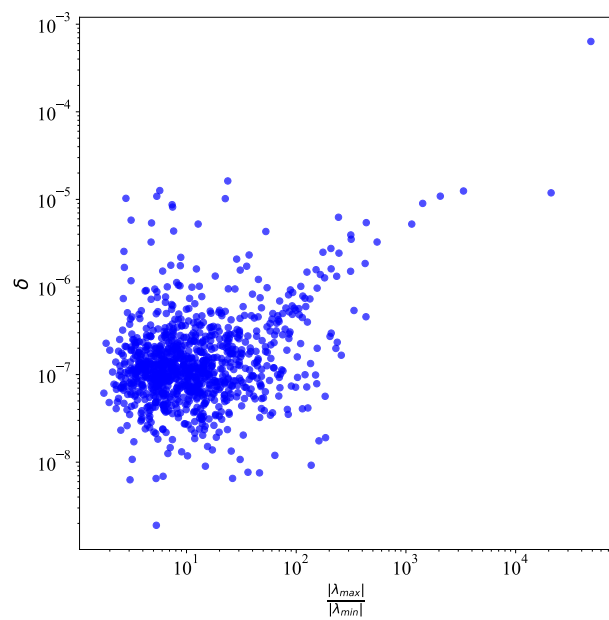


Рис. 17. Влияние соотношения максимального и минимального собственных чисел матрицы на вычислительную устойчивость алгоритмов для матриц 3-х диагонального вида

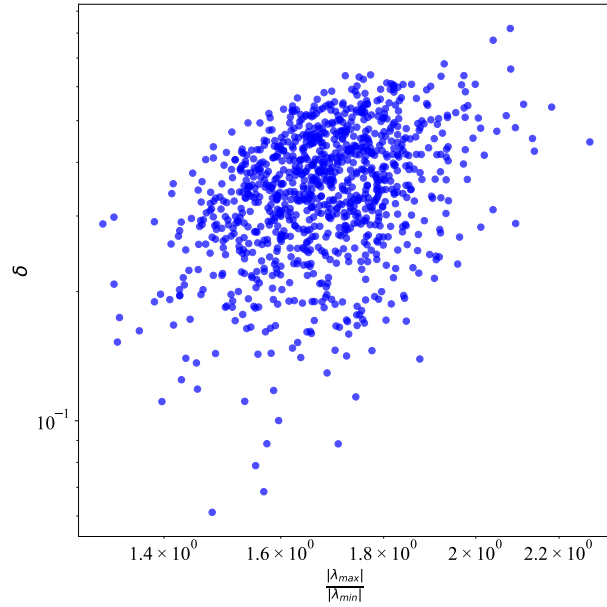


Рис. 18. Влияние соотношения максимального и минимального собственных чисел матрицы на вычислительную устойчивость алгоритмов для положительно-определённых матриц

Высокое значение отношения максимального по модулю собственного числа к минимальному по модулю собственному числу, или, что эквивалентно, высокое число обусловленности матрицы, может существенно влиять на вычислительную устойчивость алгоритмов линейной алгебры, увеличивая вероятность накопления ошибок округления и потери точности в решениях.

7 Исследование влияния числа обусловленности на вычислительную устойчивость решения СЛАУ

Для исследования влияния числа обусловленности на вычислительную устойчивость алгоритмов построены графики $\delta(\frac{|\lambda_{max}|}{|\lambda_{min}|})$ для всех видов рассмотренных матриц (рисунки 19-21).

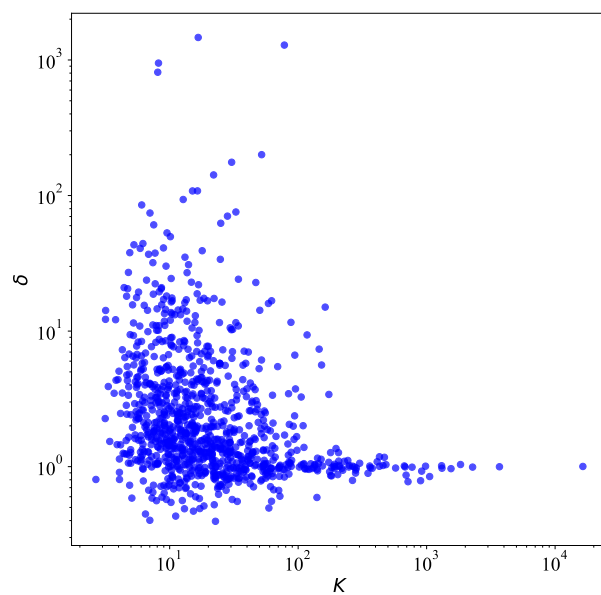


Рис. 19. Влияние числа обусловленности матрицы на вычислительную устойчивость алгоритмов для матриц общего вида

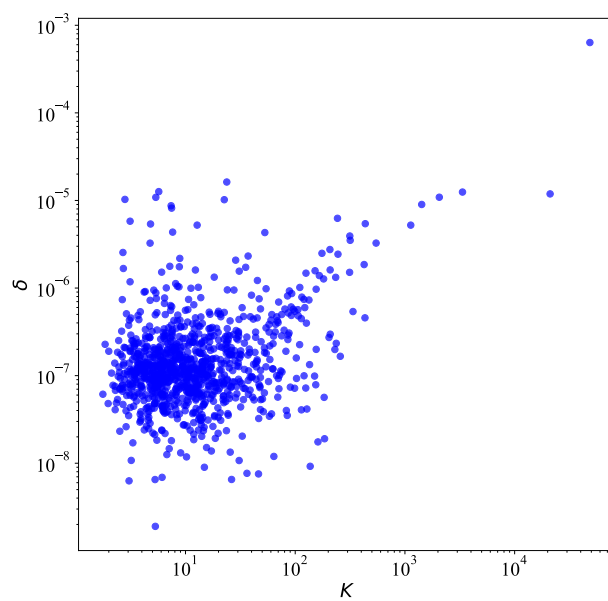


Рис. 20. Влияние числа обусловленности матрицы на вычислительную устойчивость алгоритмов для матриц 3-х диагонального вида

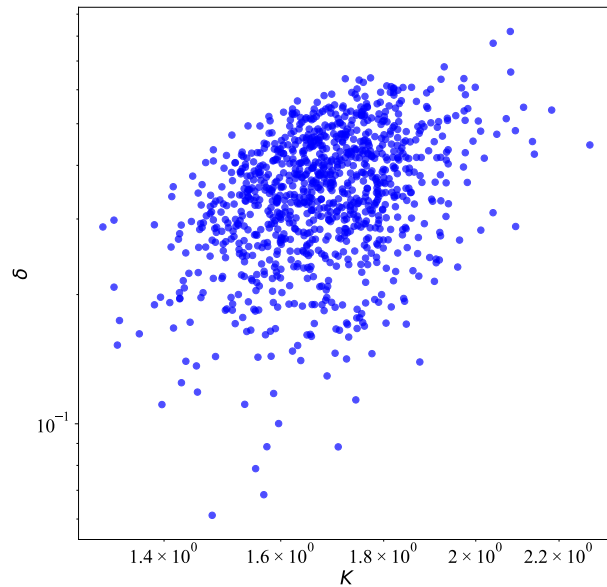


Рис. 21. Влияние числа обусловленности матрицы на вычислительную устойчивость алгоритмов для положительно-определённых матриц

Высокое число обусловленности $K(\mathbf{A})$ матрицы является предупреждающим знаком о потенциальной численной нестабильности и увеличенном риске значительных ошибок в результатах алгоритмов линейной алгебры, таких как методы Гаусса и Холецкого. Это особенно важно учитывать при решении систем линейных уравнений, где исходные данные могут содержать ошибки или быть подвержены ошибкам округления.

Заключение

В ходе лабораторной работы были выполнены следующие задачи:

1. Разработана функция, которая возвращает решение СЛАУ, полученное с помощью метода Гаусса с частичным выбором главного элемента и без него (листинг 1).
2. Разработана функция, которая возвращает решение СЛАУ, полученное с помощью метода прогонки (листинг 2).
3. В качестве универсального метода выбран метод Гаусса с выбором главного элемента.
4. Разработан алгоритм генерации случайных невырожденных матриц (листинг 3).

5. Проведенный эксперимент с 1000 случайными матрицами каждого типа и решением СЛАУ $A^{(j)}x = [1, 1, 1, 1, 1, 1]^T$ с использованием "универсального" и "специального" методов показал различия в относительных погрешностях, выраженных через среднеквадратичную и супремум-нормы, что демонстрируется на гистограммах, представленных на рисунках 1 и 2.
6. Разработан алгоритм генерации случайных положительно-определённых матриц (листинг 4).
7. Разработана функция, которая возвращает решение СЛАУ, полученное с помощью разложения Холецкого (листинг 5).
8. Проведён эксперимент с 1000 случайными матрицами каждого типа и решением СЛАУ $A^{(j)}x = [1, 1, 1, 1, 1, 1]^T$ с использованием "универсального" метода и разложением Холецкого показал различия в относительных погрешностях, выраженных через среднеквадратичную и супремум-нормы, что демонстрируется на гистограммах, представленных на рисунках 5 и 6.
9. Для всех рассмотренных ранее матриц построены гистограммы распределения спектральных радиусов и распределения чисел обусловленности (рисунки 7-12).
10. Проанализировано влияние значения спектрального радиуса матрицы на вычислительную устойчивость рассмотренных алгоритмов (рисунки 13-15).
11. Проанализировано влияние значения отношения максимального по модулю собственного числа к минимальному по модулю собственному числу на вычислительную устойчивость алгоритма (рисунки 16-18).
12. Проанализировано влияние числа обусловленности на вычислительную устойчивость алгоритма (рисунки 19-21).

Список использованных источников

1. Першин А.Ю. Лекции по курсу «Вычислительная математика». Москва, 2018-2021. С. 42–71. URL: <https://archrk6.bmstu.ru/index.php/f/810046>.
2. Соколов, А.П. Инструкция по выполнению лабораторных работ (общая). Москва: Соколов, А.П., 2018-2021. С. 9. URL: <https://archrk6.bmstu.ru>. (облачный сервис кафедры РК6).
3. Першин А.Ю., Соколов А.П., Гудым А.В. Сборник постановок задач на лабораторные работы по курсу «Вычислительная математика»: Учебное пособие. [Электронный ресурс]. Москва, 2023. С. 24–25. URL: <https://archrk6.bmstu.ru>. (облачный сервис кафедры РК6).

Выходные данные

Рогалева Ю.А. Отчет о выполнении лабораторной работы №4 по дисциплине «Вычислительная математика». [Электронный ресурс] — Москва: 2023. — 29 с. URL: <https://sa2systems.ru:88> (система контроля версий кафедры РК6)

Постановка:



аспирант кафедры РК-6, Гудым А.В.

Решение и вёрстка:



студент группы РК6-52Б, Рогалева Ю.А.

2023, осенний семестр