



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

по дисциплине «Вычислительная математика»

Студент:	Колыхалов Дмитрий Витальевич
Группа:	РК6-54Б
Тип задания:	лабораторная работа
Тема:	Устойчивость прямых методов решения СЛАУ

Студент

подпись, дата

Колыхалов Д.В.

Фамилия, И.О.

Преподаватель

подпись, дата

Фамилия, И.О.

Москва, 2023

Содержание

Устойчивость прямых методов решения СЛАУ	3
Задание	3
Цель выполнения лабораторной работы	4
1 Метод Гаусса	5
2 Метод прогонки	7
3 Выбор "универсального" метода	9
4 Генерация случайных матриц	9
5 Анализ решения СЛАУ разными методами	10
6 Генерация положительно определенных матриц	13
7 Разложение Холецкого	14
8 Анализ метода Холецкого	15
9 Распределение спектральных радиусов и чисел обусловленности	16
10 Влияние спектрального радиуса матрицы на вычислительную устойчи- вость	20
11 Влияние значение отношения максимального по модулю собственного числа к минимальному по модулю собственному числу матрицы на вы- числительную устойчивость	23
12 Влияние числа обусловленности матрицы на вычислительную устойчи- вость	25
Заключение	27

Устойчивость прямых методов решения СЛАУ

Задание

Решение систем линейных алгебраических уравнений (СЛАУ) - ключевой этап огромного множества задач 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 рассматриваются матрицы нескольких видов и с помощью генерации большого количества случайных матриц демонстрируется наличие или отсутствие вычислительной неустойчивости у метода Гаусса, метода прогонки и разложения Холецкого.

Задача 4.3 (устойчивость прямых методов решения СЛАУ)

Требуется (базовая часть):

1. Написать функцию `gauss(A, b, pivoting)`, которая возвращает решение СЛАУ $Ax = b$, полученное с помощью метода Гаусса. Если параметр `pivoting=True`, то решение должно находиться с частичным выбором главного элемента. Если `pivoting=False`, то выбора главного 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 погрешности 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.

- (с) Является ли выбранный “специальный” метод вычислительно устойчивым? Почему?

Требуется (продвинутая часть):

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

Требуется (опциональное задание):

13. Написать функцию `iter_refinement(A, b)`, которая возвращает решение СЛАУ $Ax = b$, полученное с помощью итерационного уточнения, полученного на основе решения методом Гаусса с частичным выбором главного элемента.
14. Сравнить данный метод решения с “универсальным” (оценить разницу решений) для всех приведенных ранее матриц.

Цель выполнения лабораторной работы

Цель выполнения лабораторной работы: разработать генератор для создания случайной трехдиагональной, положительно-определенной матрицы и общего вида. Реализовать методы решения СЛАУ на основе метода Гаусса, прогонки и разложения Холецкого. Сравнить методы решения. Выявить зависимость вычислительной устойчивости от характеристик матрицы.

1 Метод Гаусса

Метод Гаусса является базовым методом решения СЛАУ и состоит в приведении матрицы A к треугольному виду с помощью элементарных преобразований. Пример расширенной матрицы представлен ниже:

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

Для того, чтобы привести матрицу A к треугольному виду, необходимо последовательно обнулять элементы, находящиеся под главной диагональю. Так, если из строк $i = 2, 3, \dots, n$ отнять первую строку, домноженную на $\frac{a_{i1}}{a_{11}}$, то все элементы, расположенные под a_{11} будут равны нулю. Обозначив это элементарное преобразование как:

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

где i обозначает i -ю строку. Тогда, применив операцию к расширенной матрицы, получается:

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

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:

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

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

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

Этот этап метода Гаусса называется прямым ходом. Очевидно, что тот же метод можно использовать для получения нижней треугольной матрицы. Было замечено, что решение СЛАУ, в которой матрица имеет треугольную форму, легко находится с помощью так называемого обратного хода метода Гаусса:

$$x_n = \frac{b_n^{(n-1)}}{a_{n,n}^{(n-1)}} \quad (6)$$


```

24         n = Matrix[j][i]
25         Matrix[j][i] = Matrix[j][i] - (Matrix[i][i] * Matrix[j][i]) / Matrix[i][i]
26         free[j] = free[j] - (free[i] * n) / Matrix[i][i]
27         for k in range(i + 1, length):
28             Matrix[j][k] = Matrix[j][k] - (n * Matrix[i][k]) / Matrix[i][i]
29     else:
30         for i in range(length):
31             for j in range(i + 1, length):
32                 n = Matrix[j][i]
33                 Matrix[j][i] = Matrix[j][i] - Matrix[i][i] * Matrix[j][i] / Matrix[i][i]
34                 free[j] = free[j] - (n * free[i]) / Matrix[i][i]
35                 for k in range(i + 1, length):
36                     Matrix[j][k] = Matrix[j][k] - (n * Matrix[i][k]) / Matrix[i][i]
37     solution=[]*length
38     solution[length-1]= free[length - 1] / Matrix[length - 1][length - 1]
39     for i in range(length-2,-1,-1):
40         n=0
41         for j in range(i+1,length):
42             n= n + solution[j] * Matrix[i][j]
43         solution[i]= (free[i] - n) / Matrix[i][i]
44     return solution

```

2 Метод прогонки

Метод прогонки - специальный метод решения СЛАУ, имеющих трехдиагональную матрицу. Пусть имеется следующее матричное уравнение:

$$\begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & \cdots & \cdots & 0 \\ a_{21} & a_{22} & a_{23} & \cdots & \cdots & \cdots & 0 \\ 0 & a_{32} & a_{33} & a_{34} & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & \ddots & a_{n-1,n-2} & a_{n-1,n-1} & a_{n-1,n} \\ 0 & \cdots & \cdots & \cdots & 0 & a_{n,n-1} & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix} \quad (9)$$

где каждая строка эквивалентна рекуррентному соотношению вида:

$$a_{i,i-1}x_{i-1} + a_{ii}x_i + a_{i,i+1}x_{i+1} = b_i \quad (10)$$

Очевидно, что методом последовательного исключения можно привести такую СЛАУ

к верхней треугольной форме:

$$\begin{bmatrix} \tilde{a}_{11} & \tilde{a}_{12} & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \tilde{a}_{22} & \tilde{a}_{23} & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & a_{33} & a_{34} & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 & 0 \\ \vdots & & & \ddots & 0 & \tilde{a}_{n-1,n-1} & \tilde{a}_{n-1,n} \\ 0 & \cdots & \cdots & \cdots & 0 & 0 & \tilde{a}_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \tilde{b}_3 \\ \vdots \\ \vdots \\ \tilde{b}_{n-1} \\ \tilde{b}_n \end{bmatrix} \quad (11)$$

Тогда решение для x_{i-1} может быть выражено через x_i :

$$x_{i-1} = \frac{\tilde{b}_{i-1} - \tilde{a}_{i-1,i}x_i}{\tilde{a}_{i-1,i-1}} \quad (12)$$

что задает выражение для обратного хода метода Гаусса. Вместо явного вывода неизвестных коэффициентов, строится рекуррентное соотношение, позволяющее рекурсивно найти нужные коэффициенты. Для упрощения записи переопределяют коэффициенты:

$$x_{i-1} = \gamma_i x_i + \beta_i \quad (13)$$

Тогда подстановка в (10) даст:

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

Сравнивая формулы (13) и (14) были выражены коэффициенты γ, β :

$$\gamma_{i+1} = \frac{-a_{i,i+1}}{a_{i,i-1}\gamma_i + a_{ii}} \quad (15)$$

$$\beta + i + 1 = \frac{b_i - a_{i,i-1}\beta_i}{a_{i,i-1}\gamma_i + a_{ii}} \quad (16)$$

Последовательно вычислив все коэффициенты γ_i и β_i , решение СЛАУ находится с помощью обратного хода метода Гаусса по формуле (13). Для завершения построения метода прогонки достаточно найти выражения для коэффициентов γ_1 и β_1 и неизвестной x_n .

$$\gamma_1 = \frac{-a_{12}}{a_{11}} \quad (17)$$

$$\beta_1 = \frac{b_1}{a_{11}} \quad (18)$$

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

Функция, реализующая метод прогонки для трехдиагональных матриц, представлена в листинге (2).


```

1 def thomas(A, b):
2     Matrix=np.array(A, dtype=np.float32)
3     length = len(Matrix[0])
4     free = np.array(b)
5     gamma=[]*length
6     nu=[]*length
7     solution=[]*length
8     gamma[0]= -1*Matrix[0][1] / (Matrix[0][0])
9     nu[0]= free[0] / Matrix[0][0]
10    for i in range(1,length-1):
11        znam=(Matrix[i][i] + Matrix[i][i - 1] * gamma[i - 1])
12        gamma[i]= -1*Matrix[i][i + 1] / znam
13        nu[i]= (free[i] - nu[i - 1] * Matrix[i][i - 1]) / znam
14    gamma[length-1]=0
15    znam = (Matrix[length - 1][length - 1] + Matrix[length - 1][length - 2] *
16            gamma[length - 2])
17    nu[length-1] = (free[length - 1] - nu[length - 2] * Matrix[length - 1][length - 2]) /
18                    znam
19    solution[length-1]=nu[length-1]
20    for i in range(length-2,-1,-1):
21        solution[i]=(gamma[i]*solution[i+1])+nu[i]
22    return solution

```

3 Выбор "универсального" метода

В случае квадратных матриц общего вида, могут сравниваться только два реализованных метода, а именно метод Гаусса с частичным выбором главного элемента и без него. Метод прогонки используется только для трехдиагональных матриц, следовательно он не подходит.

Если элемент $a_{kk}^{(k)}$ мал по сравнению с $a_{ik}^{(k)}$, множитель $m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$ будет иметь большое значение, что приводит к усилению погрешности округления в результате ряда последующих умножений на m_{ik} . Более того, деление на малый элемент $a_{kk}^{(k)}$ происходит также во время обратного хода Гаусса и усиливает накопленные погрешности в числителе. При использовании метода Гаусса с частичным выбором главного элемента, на диагональный элемент ставится наибольший по модулю элемент, относительно которого происходит вычисление последующих, следовательно погрешность вычисления минимизируется. Из вышеизложенного следует, что метод Гаусса с частичным выбором главного элемента является "универсальным".

4 Генерация случайных матриц

Матрица порядка n является невырожденной, если выполняется любое из перечис-

ленных условий:

- матрица диагональная с ненулевыми диагональными элементами
- матрица верхняя треугольная с ненулевыми диагональными элементами
- матрица нижняя треугольная с ненулевыми диагональными элементами
- матрица унитреугольная (т.е. верхние треугольные матрицы у которых диагональные элементы равны 1)

Для генерации невырожденных матриц общего и треугольного вида за основу бралась единичная матрица, размерность которой 6. Путем элементарных преобразований, т.е. умножение строк на число и сложение строк матрицы, единичная матрица преобразовывалась в требуемую.

Для получения матрицы общего вида все строки единичной матрицы необходимо умножить на случайное число $|k| < 1$ и прибавить к первой строке, для того, чтобы заполнить все ее элементы. При одной полностью заполненной строке матрицы, появляется возможность заполнить все оставшиеся элементы, путем сложения первой строки со всеми остальными. Чтобы элементы матрицы были случайны при каждом итерации сложения, строка умножается на случайное число $|k| < 1$.

Для получения трехдиагональной матрицы, все строки единичной матрицы необходимо умножить на случайное число $|k| < 1$ и последовательно складывать текущую строку со следующей по следующему правилу $a_i + a_{i+1}$ при $i = 1, \dots, n-1$ так заполняется нижняя диагональ. После чего, складывать получившиеся строки как $a_i + a_{i-1}$ при $i = n, \dots, 2$ для заполнения верхней диагонали. Чтобы элементы матрицы были случайны при каждой итерации сложения, строка умножается на случайное число $|k| < 1$.

5 Анализ решения СЛАУ разными методами

Сгенерировав 1000 матриц каждого типа, необходимо выбрать "специальный" метод по количеству арифметико-логических операций. Для матрицы общего вида - им будет являться метод Гаусса без частичного выбора главного элемента, поскольку в противном случае, придется находить подходящий элемент на главную диагональ, что увеличит количество операций.

Для трехдиагональной матрицы оптимальным будет являться метод прогонки, который может сводиться к $O(n)$, что заметно меньше $O(n^3)$ метода Гаусса.

Сравнивать найденные "специальные" методы с "универсальными" необходимо по относительной погрешности вычисления с использованием среднеквадратичной и супремум-нормы.

Формула относительной погрешности:

$$\frac{|x - \tilde{x}|}{|x|} \quad (20)$$

где x - решение найденное "универсальным" методом, \tilde{x} - решение найденное "специальным" методом.

Формула среднеквадратичной нормы:

$$\|x\|_2 = \left(\sum_{i=1}^n x_i^2 \right)^{\frac{1}{2}} \quad (21)$$

Формула супремум-нормы:

$$\|x\|_{\infty} = \max_{i \in [1;n]} |x_i| \quad (22)$$

Результат сравнения "специального" метода с "универсальным" для матрицы общего вида представлен на рисунках (1) и (2).

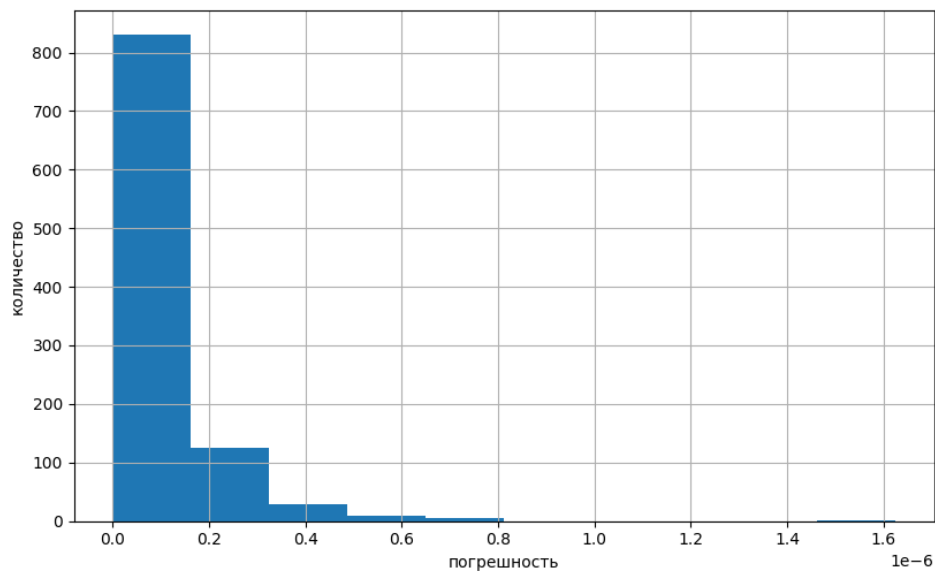


Рис. 1. Гистограмма распределения относительной погрешности для метода Гаусса с использованием среднеквадратичной нормы

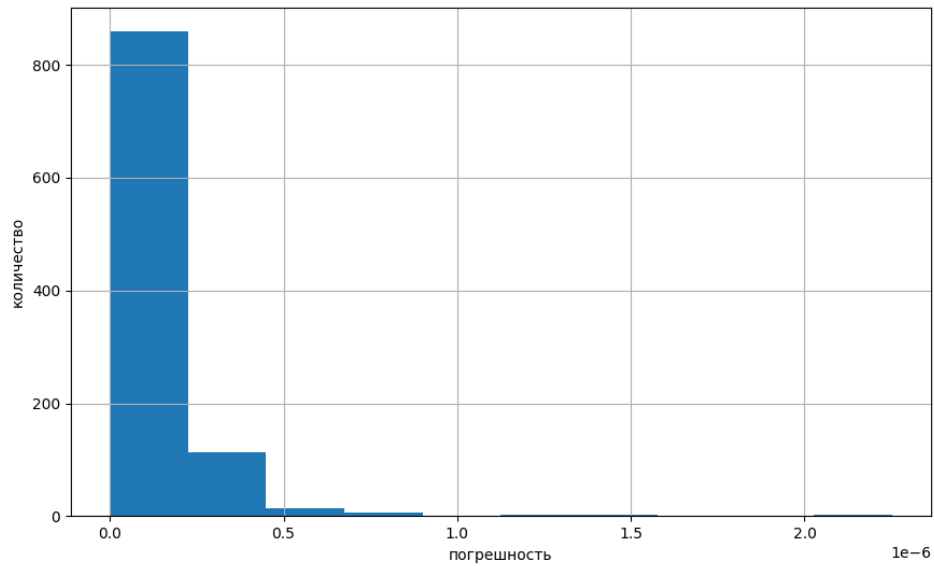


Рис. 2. Гистограмма распределения относительной погрешности для метода Гаусса с использованием супремум нормы

Результат сравнения "специального" метода с "универсальным" для матрицы трех-диагонального вида представлен на рисунках (3) и (4).

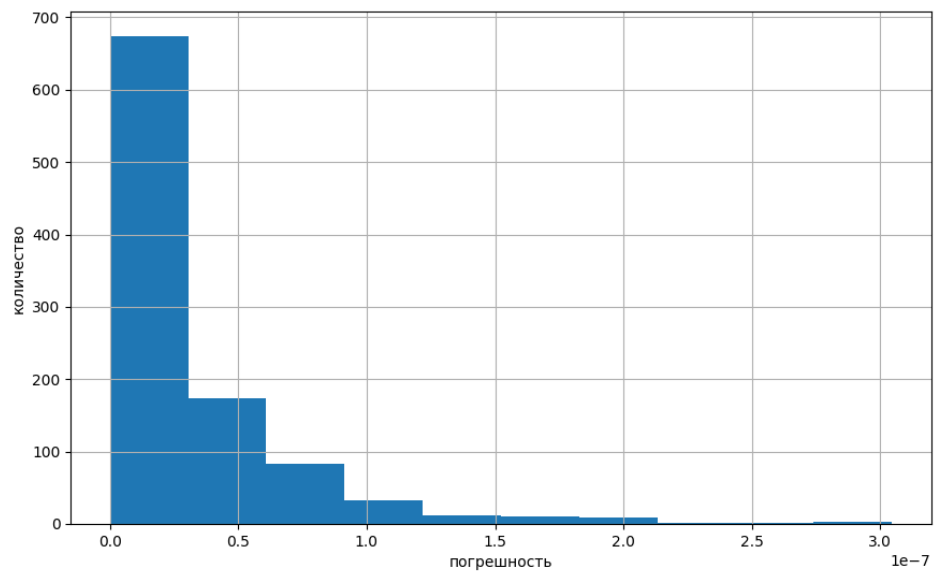


Рис. 3. Гистограмма распределения относительной погрешности для метода прогонки с использованием среднеквадратичной нормы

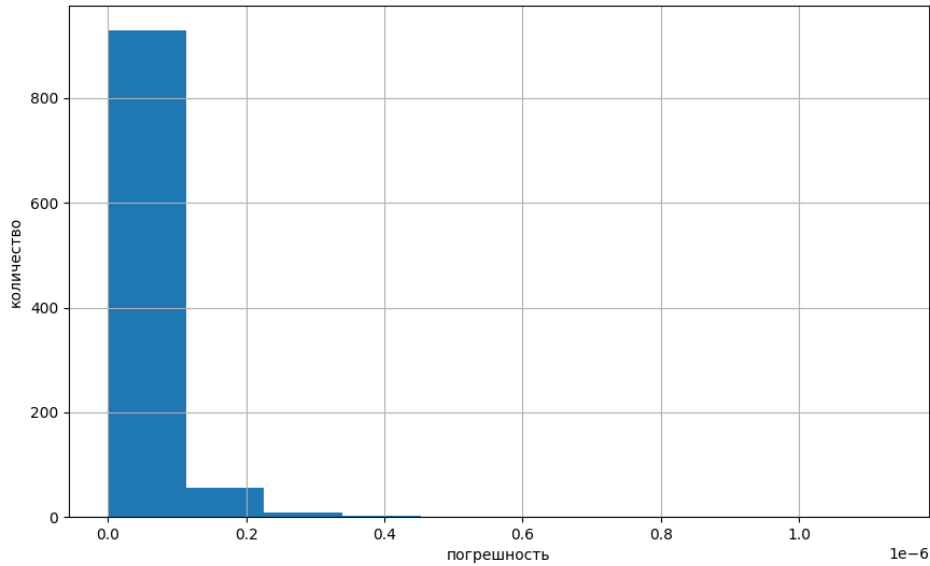


Рис. 4. Гистограмма распределения относительной погрешности для метода прогонки с использованием супремум нормы

По представленным гистограммам можно сделать вывод о том, что метод Гаусса, как и метод прогонки является вычислительно устойчивым, т.к большая часть погрешности сосредоточена на одном участке вблизи нуля. При этом на рисунках видно, что хоть разброс и небольшой, но он **присутствует**. Для метода Гаусса некоторые погрешности достигают $2 \cdot 10^{-6}$, для метода прогонки $3 \cdot 10^{-7}$.

6 Генерация положительно определенных матриц

Матрица A называется положительно определенной, если она симметричная, и верным является неравенство $x^T A x > 0$ для любого вектора $x \neq 0$ подходящей размерности. Положительно определенная матрица обладает следующим свойством: $A = LL^T$, где L - нижняя треугольная матрица с ненулевыми элементами на диагонали.

Для получения положительно определенной матрицы, все строки единичной матрицы необходимо умножить на случайное число $|k| < 1$ и последовательно складывать строки по следующему правилу $a_i + a_{i-1}$, при $i = 2, \dots, n$. В результате чего получается заполненная нижняя треугольная матрица. Чтобы элементы матрицы были случайны, при каждой итерации, строка умножается на случайное число $|k| < 1$. После чего полученную матрицу необходимо умножить на транспонированную матрицу L^T , в результате чего и генерируется случайная положительно определенная матрица.

7 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.

Неизвестные элементы матрицы L вычисляются следующим образом:

$$l_{ii} = \sqrt{a_{ii} - \sum_{j=1}^{i-1} l_{ij}^2}, i = 1, \dots, n \quad (23)$$

$$l_{ij} = \frac{1}{l_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} \right), j < i \quad (24)$$

В результате разложение получается нижняя треугольная матрица L из которой можно получить верхнюю треугольную матрицу, транспонировав L .

В СЛАУ $Ax=b$ решение x получается последовательным решением двух треугольных систем уравнений: $Ly = b$ и $L^T x = y$.

Код, реализующий разложение Холецкого представлен в листинге (3).

Листинг 3. Программная реализация метода решения СЛАУ через разложение Холецкого

```
1 def cholesky(A, b):
2     Matrix=np.array(A, dtype=np.float32)
3     length = len(Matrix[0])
4     b=np.array(b)
5     L=[[0]*length]*length
6     L=np.array(L,dtype=np.float32)
7     for i in range(length):
8         for j in range(i+1):
9             if i==j:
10                n=0
11                for k in range(i):
12                    n=n+pow(L[i][k],2)
13                L[i][j]= math.sqrt(Matrix[i][i] - n)
14            else:
15                n=0
16                for k in range(j):
17                    n+=L[i][k]*L[j][k]
18                L[i][j]= (Matrix[i][j] - n) / L[j][j]
19     L_transp=trnsnp(L)
20     y = [] * length
21     y[0]=b[0]/L[0][0]
22     for i in range(1,length):
23         n =0
24         for j in range(i):
25             n+=y[j]*L[i][j]
26         y[i]=(b[i]-n)/L[i][i]
```

[set] * (None) * (None) * (None), (None), (None))

```

27  solution = [] * length
28  solution[length - 1] = y[length - 1] / L_transp[length - 1][length - 1]
29  for i in range(length-2,-1,-1):
30      n=0
31      for j in range(i+1,length):
32          n=n+solution[j]*L_transp[i][j]
33      solution[i]=(y[i]-n)/L_transp[i][i]
34  return solution

```

8 Анализ метода Холецкого

Для сравнения метода Холецкого с "универсальным" методом, были сгенерированы 1000 положительно определенных матриц. Сравнения были произведены по относительной погрешности вычисления с использованием среднеквадратичной и супремум-нормы.

Гистограммы, полученные для каждой нормы представлены на рисунках (5) и (6).

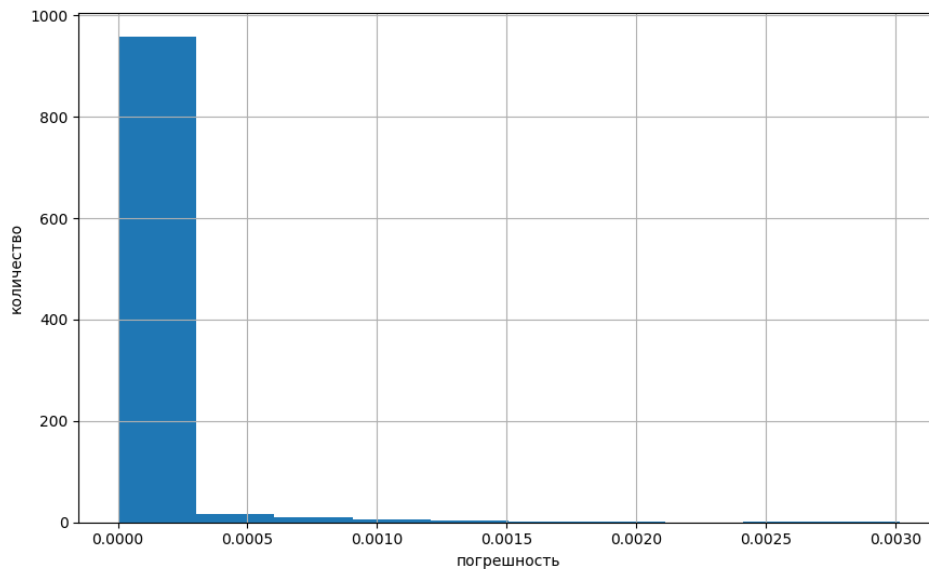


Рис. 5. Гистограмма распределения относительной погрешности для метода Холецкого с использованием среднеквадратичной нормы

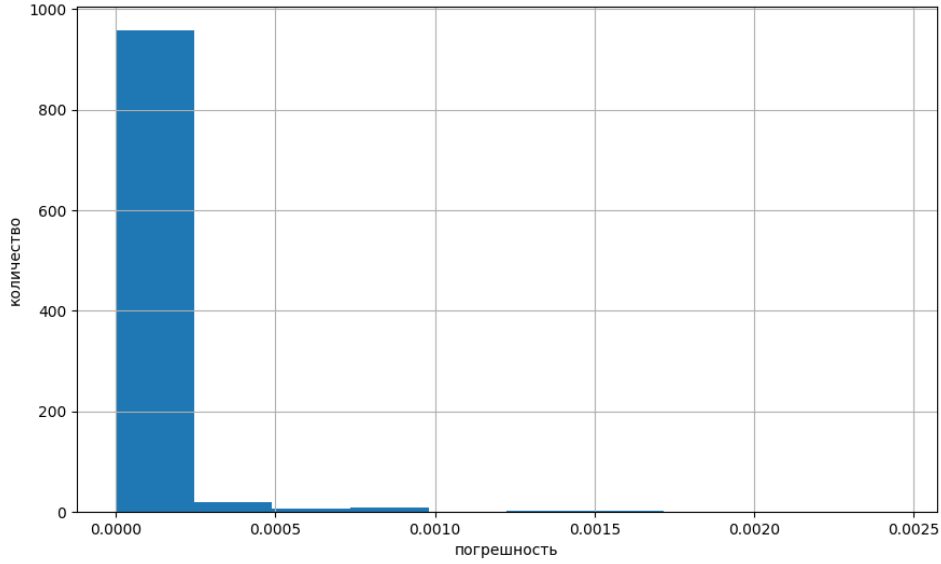


Рис. 6. Гистограмма распределения относительной погрешности для метода Холецкого с использованием супремум нормы

По представленным гистограммам можно сделать вывод о том, что метод Холецкого является вычислительно устойчивым, т.к. большая часть погрешности сосредоточена на одном участке вблизи нуля. При этом на рисунках видно, что величина погрешности заметно увеличилась по сравнению с методом Гаусса и методом прогонки. Это может объясняться тем, что при вычислении элементов нижней треугольной матрицы использовались операции возведение в степень и извлечение корня, что для чисел с плавающей точкой не всегда дает точный результат.

9 Распределение спектральных радиусов и чисел обусловленности

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

$$\rho(A) = \max_{i \in [1; m]} |\lambda_i|, \quad (25)$$

где λ_i – одно из m собственных чисел матрицы A .

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

$$\|x - \tilde{x}\| \leq \|r\| \cdot \|A^{-1}\| \quad (26)$$

и при $x, b \neq 0$:

$$\frac{\|x - \tilde{x}\|}{\|x\|} \leq K(A) \frac{\|r\|}{\|b\|} \quad (27)$$

где $K(A) = \|A\| \cdot \|A - I\|$ называется числом обусловленности.

Для нахождения спектральных радиусов была использована функция `numpy.linalg.eigvals()`, которая возвращает массив собственных чисел, среди которых необходимо найти максимальное по модулю.

Для нахождения чисел обусловленности была использована функция `numpy.linalg.cond()`, которая возвращает число обусловленности в виде значения с плавающей точкой.

Гистограммы, полученные для матрицы общего вида, представлены на рисунках (7) и (8).

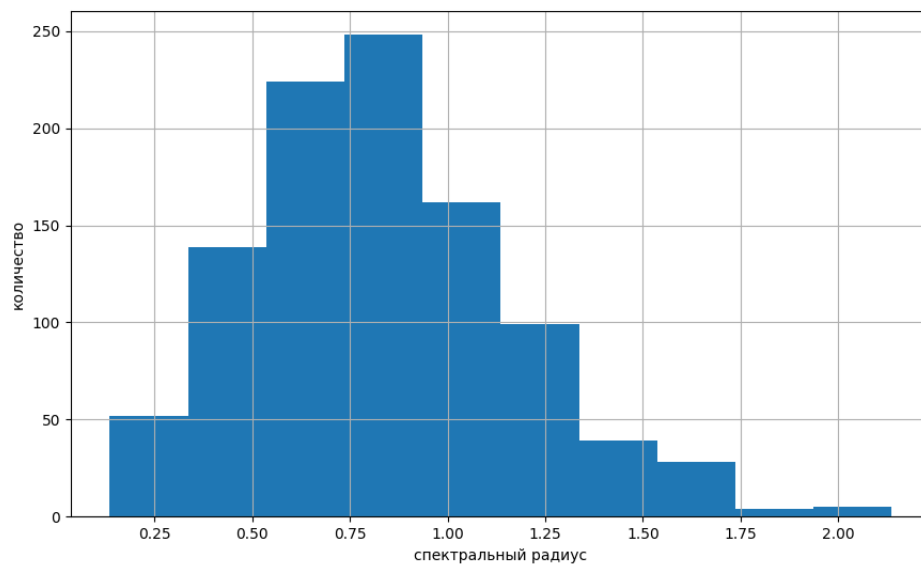


Рис. 7. Гистограмма распределения спектрального радиуса для матрицы общего вида

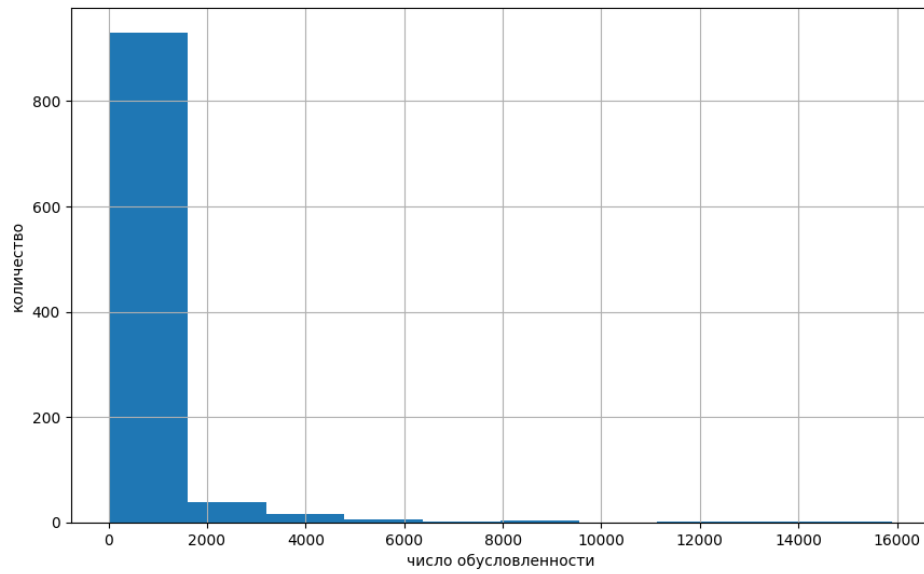


Рис. 8. Гистограмма распределения чисел обусловленности для матрицы общего вида

Гистограммы, полученные для трехдиагональной матрицы представлены на рисунках (9) и (10).

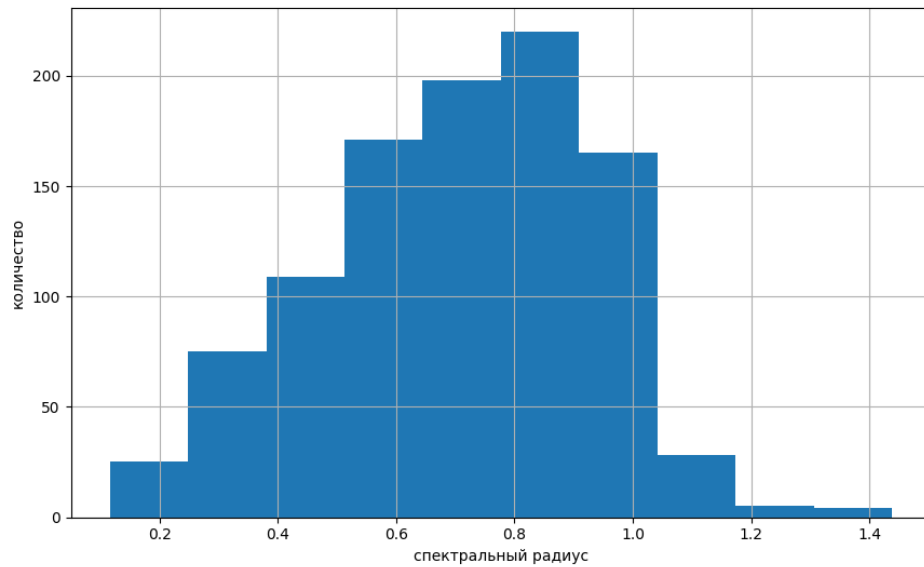


Рис. 9. Гистограмма распределения спектрального радиуса для трехдиагональной матрицы

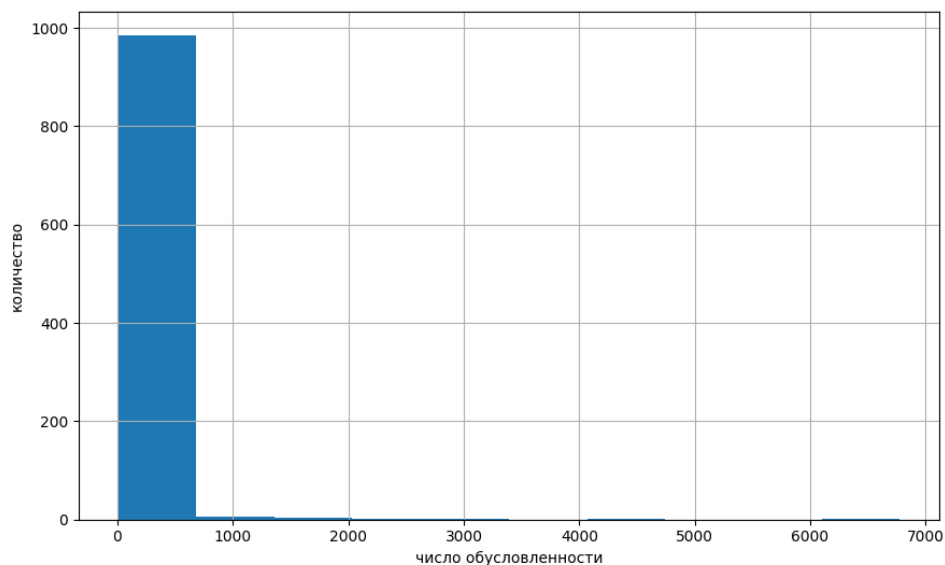


Рис. 10. Гистограмма распределения чисел обусловленности для трехдиагональной матрицы

Гистограммы, полученные для положительно определенной матрицы представлены на рисунках (11) и (12).

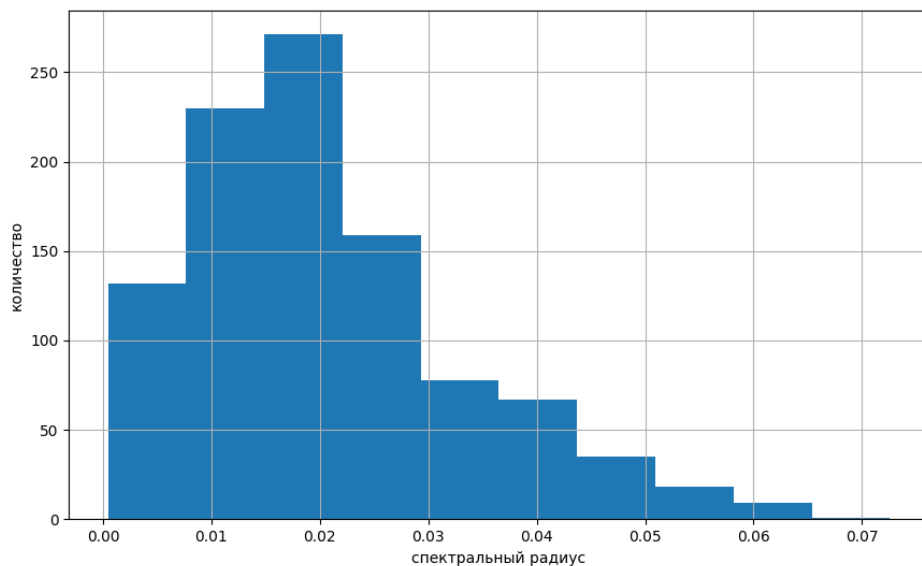


Рис. 11. Гистограмма распределения спектрального радиуса для положительно определенной матрицы

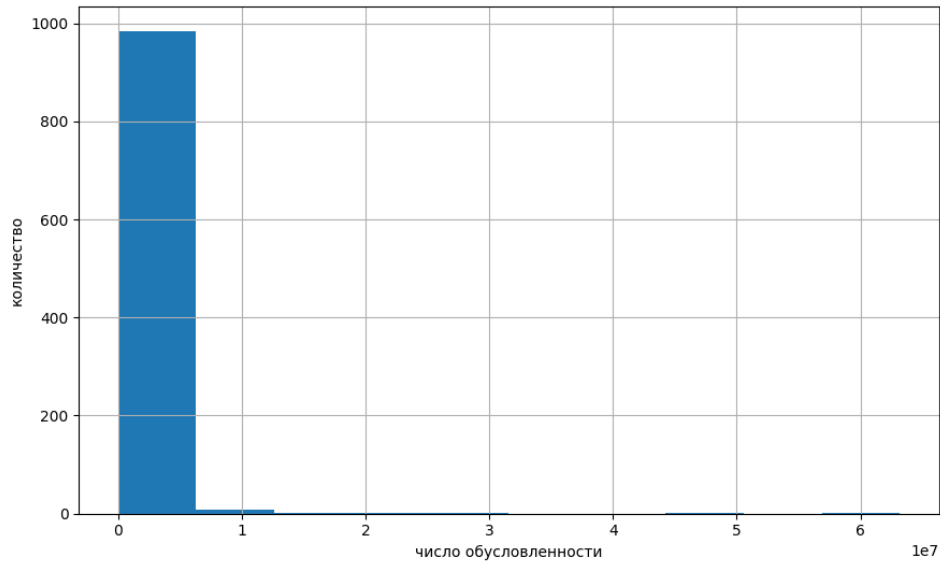


Рис. 12. Гистограмма распределения чисел обусловленности для положительно определенной матрицы

По полученным результатам можно сделать вывод о том, что распределение спектральных радиусов для всех трех матриц имеет нормальный вид, т.е. относится к нормальному распределению. Распределение чисел обусловленности для каждой матрицы имеет свой порядок, для матрицы общего вида основной объем расположен между 0 и 2000, хотя может достигать и 14000. Для трехдиагональной матрицы порядок уменьшается вдвое, для положительно определенной матрицы основной объем расположен между 0 и 10^7 , редко превышая $6 \cdot 10^7$.

10 Влияние спектрального радиуса матрицы на вычислительную устойчивость

Для того, чтобы оценить влияние спектрального радиуса матрицы на вычислительную устойчивость необходимо выявить есть ли зависимость между погрешностью решения и спектральным радиусом. Для этого, необходимо построить точечный график и проследить как соотносятся между собой две характеристики.

Графики для каждой матрицы представлены на рисунках (13), (14) и (15).

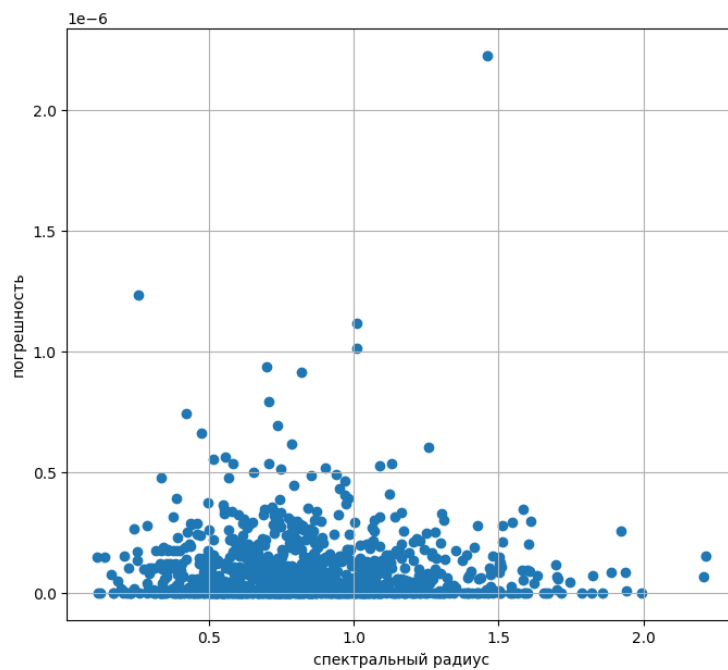


Рис. 13. График зависимости погрешности решения от спектрального радиуса для матрицы общего вида

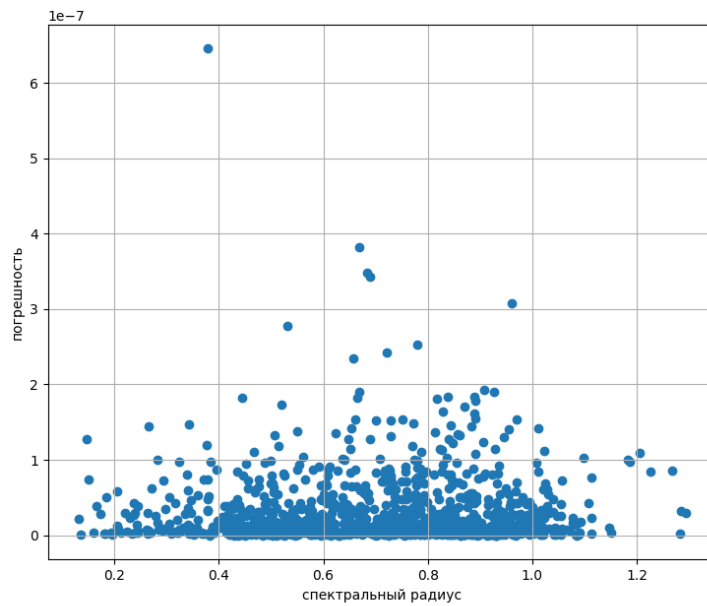


Рис. 14. График зависимости погрешности решения от спектрального радиуса для трехдиагональной матрицы

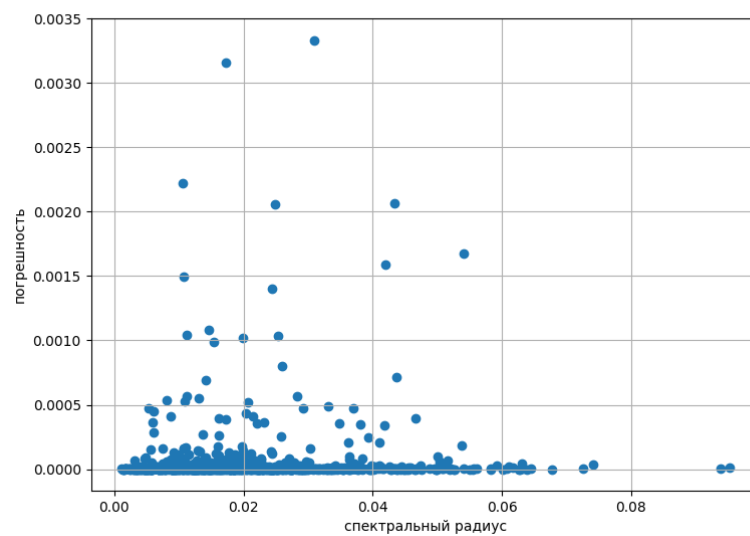


Рис. 15. График зависимости погрешности решения от спектрального радиуса для положительно определенной матрицы

По полученным графикам можно сделать вывод о том, что чем меньше порядок

спектрального радиуса, тем точнее вычисление. При этом явной зависимости между двумя характеристиками на каждом отдельном графике не выявлено. Для трехдиагональной матрицы порядок радиуса самый малый, что влечет за собой небольшую вычислительную погрешность. Обратная картина наблюдается для положительно определенной матрицы. Следовательно, чем меньше спектральный радиус, тем метод вычислительно устойчивее.

11 Влияние значения отношения максимального по модулю собственного числа к минимальному по модулю собственному числу матрицы на вычислительную устойчивость

Для того, чтобы оценить влияние отношения максимального по модулю собственного числа к минимальному по модулю собственному числу матрицы на вычислительную устойчивость необходимо выявить есть ли зависимость между погрешностью решения и этим отношением. Для этого, необходимо построить точечный график и проследить как соотносятся между собой две эти характеристики.

Графики для каждой матрицы представлены на рисунках (16), (17) и (18).

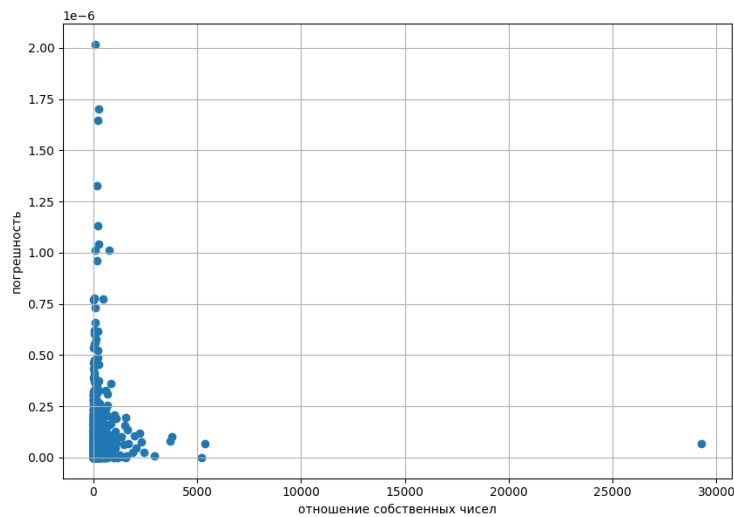


Рис. 16. График зависимости погрешности решения от отношения максимального по модулю собственного числа к минимальному по модулю собственному числу для матрицы общего вида

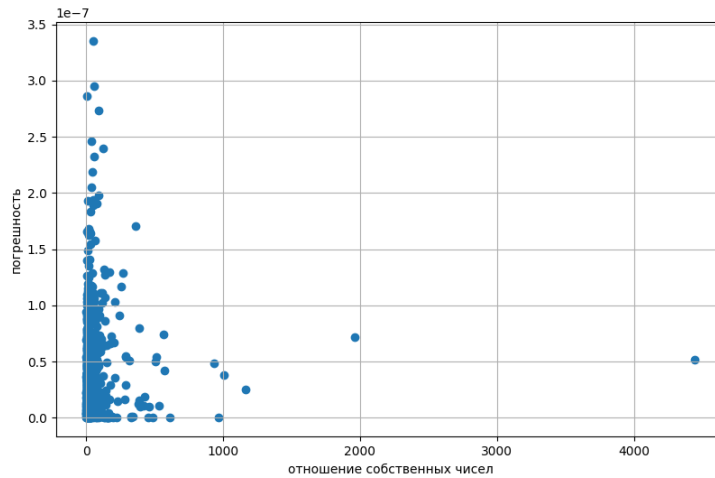


Рис. 17. График зависимости погрешности решения от отношения максимального по модулю собственного числа к минимальному по модулю собственному числу для трехдиагональной матрицы

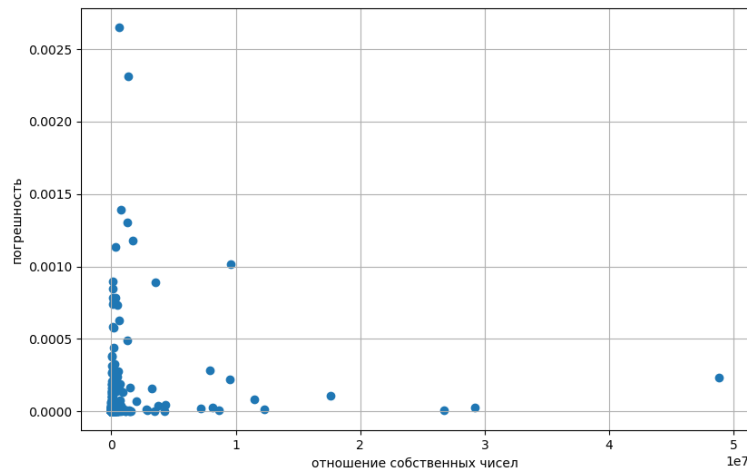


Рис. 18. График зависимости погрешности решения от отношения максимального по модулю собственного числа к минимальному по модулю собственному числу для положительно определенной матрицы

По полученным графикам можно сделать вывод о том, что чем меньше порядок отношения, тем менее точное вычисление. При этом зависимость между двумя характеристиками на каждом отдельном графике такая, что основной объем значений отношений собственных чисел расположился в небольшом промежутке, в следствие чего и наблюдается резкий всплеск погрешности. Для трехдиагональной матрицы порядок отно-

шения самый малый, что влечет за собой небольшую вычислительную погрешность. Обратная картина наблюдается для положительно определенной матрицы. Следовательно, чем больше отношение, тем метод вычислительно устойчивее.

12 Влияние числа обусловленности матрицы на вычислительную устойчивость

Для того, чтобы оценить влияние числа обусловленности матрицы на вычислительную устойчивость, необходимо выявить есть ли зависимость между погрешностью решения и числом обусловленности. Для этого, необходимо построить точечный график и проследить как соотносятся между собой две эти характеристики.

Графики для каждой матрицы представлены на рисунках (19), (20) и (21).

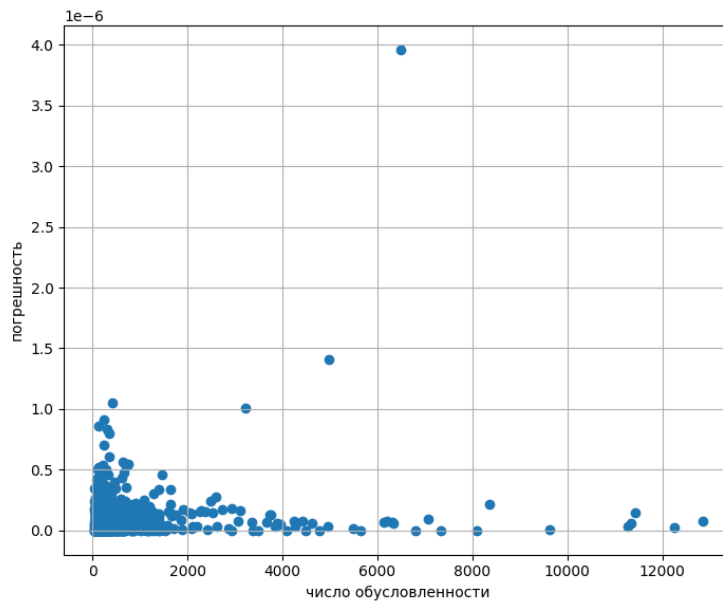


Рис. 19. График зависимости погрешности решения от числа обусловленности для матрицы общего вида

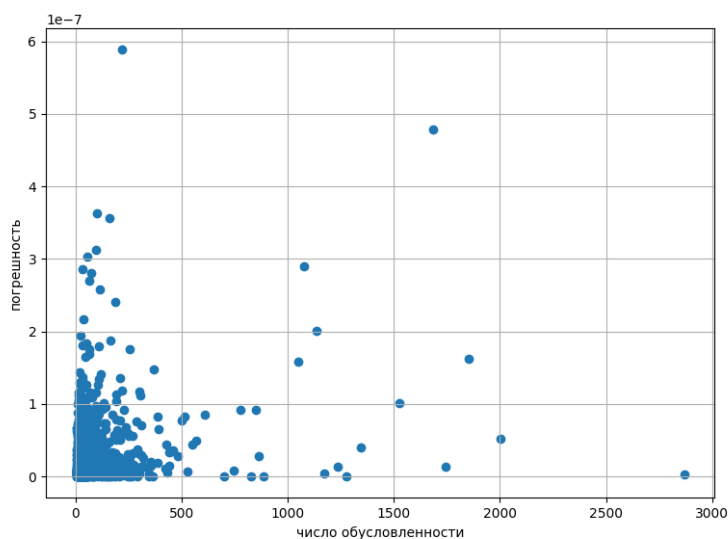


Рис. 20. График зависимости погрешности решения от числа обусловленности для трехдиагональной матрицы

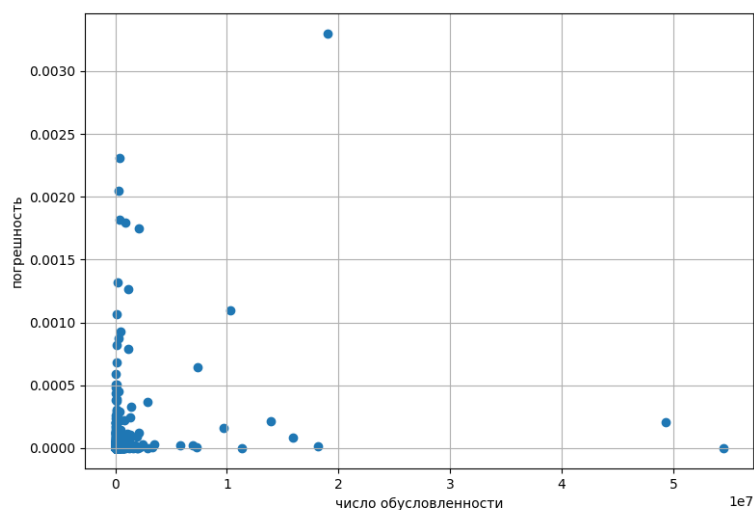


Рис. 21. График зависимости погрешности решения от числа обусловленности для положительно определенной матрицы

По полученным графикам можно сделать вывод о том, что чем меньше порядок обусловленности, тем точнее вычисление. При этом зависимость между двумя характеристиками на каждом отдельном графике такая, что основной объем значений чисел обусловленности расположился в небольшом промежутке, в следствии чего и наблюдаю-

дается резкий всплеск погрешности. Для трехдиагональной матрицы порядок обусловленности самый малый, что влечет за собой небольшую вычислительную погрешность. Обратная картина наблюдается для положительно определенной матрицы. Следовательно, чем меньше число обусловленности, тем метод вычислительно устойчивее.

Матрицы с очень большим числом обусловленности принято называть плохо обусловленными. СЛАУ с плохообусловленными матрицами требуют особого подхода.

Заключение



1. Разработаны генераторы случайных невырожденных трехдиагональных, положительно определенных матриц и общего вида.
2. Реализованы методы решения СЛАУ на основе метода Гаусса, прогонки и разложения Холецкого.
3. Выявлены зависимости между характеристиками матриц и вычислительной устойчивостью.
4. Определена вычислительная устойчивость для каждого метода решения по построенным гистограммам.

Список использованных источников

1. Першин А.Ю. Лекции по курсу «Вычислительная математика». Москва, 2018-2021. С. 140. 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. Соколов, А.П. Инструкция по выполнению заданий к семинарским занятиям (общая). Москва: Соколов, А.П., 2018-2022. С. 7. URL: <https://archrk6.bmstu.ru>. (облачный сервис кафедры РК6).
4. Першин А.Ю. Сборник задач семинарских занятий по курсу «Вычислительная математика»: Учебное пособие. / Под редакцией Соколова А.П. [Электронный ресурс]. Москва, 2018-2021. С. 20. URL: <https://archrk6.bmstu.ru>. (облачный сервис кафедры РК6).
5. Першин А.Ю., Соколов А.П. Сборник постановок задач на лабораторные работы по курсу «Вычислительная математика»: Учебное пособие. [Электронный ресурс]. Москва, 2021. С. 54. URL: <https://archrk6.bmstu.ru>. (облачный сервис кафедры РК6).

Выходные данные

Колыхалов Д.В. Отчет о выполнении лабораторной работы по дисциплине «Вычислительная математика». [Электронный ресурс] — Москва: 2023. — 28 с. URL: <https://sa2systems.ru:88> (система контроля версий кафедры РК6)

Постановка:  доцент кафедры РК-6, PhD А.Ю. Першин
Решение и вёрстка:  студент группы РК6-54Б, Колыхалов Д.В.

2023, осенний семестр