МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №7
по дисциплине «Вычислительная математика»
Тема: Исследование обусловленности задачи решения систем линейных уравнений

Студент гр. 0304	____________________	Алексеев Р.В.
Преподаватель	____________________	Попова Е.В.

Санкт-Петербург
2021

Вариант 1

Цель работы: Изучение стандартной обусловленности задач решения систем линейных уравнений при различных вариантах неточных входных данных.

Основные теоретические положения.

Рассматривается система линейных уравнений n-го порядка с вещественными коэффициентами (1)

$a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n=b_1$

$a_{21}x_1+a_{22}x_2+\ldots+a_{2n}x_n=b_2$

$\ldots$

$a_{n1}x_1+a_{n2}x_2+\ldots+a_{nn}x_n=b_n.$

В матричной форме записи эта система принимает вид (2)

$$AX=B$$

$$A=\begin{bmatrix} a_{11}\ldots a_{1n} \\ \ldots \\ a_{n1}\ldots a_{nn}\end{bmatrix}, X=\begin{bmatrix} x_1 \\ \ldots \\ x_n\end{bmatrix}, B=\begin{bmatrix} b_1 \\ \ldots \\ b_n\end{bmatrix}, \qquad (2)$$

где A – квадратная матрица коэффициентов системы, X – вектор решений системы, B – вектор свободных членов. Матрица A – невырожденная, тогда решение системы (1) существует, единственно и устойчиво по входным данным. Это означает, что задача нахождения вектора X – корректна.

Пусть $X^{¿}=\begin{bmatrix} x^{¿}{}_1 \\ \ldots \\ x^{¿}{}_n\end{bmatrix}$ – приближенное решение системы, тогда $e=X-X^{¿}$ называется вектором погрешности системы, необходимо стремиться к его уменьшению. Возможно рассматривать критерий малости вектора $r=B-AX^{¿}=B-B^{¿}=A\left(X-X^{¿}\right)$, который называется невязкой системы. Эти вектора связаны $e=X-X^{¿}=A^{-1}r$**.**

Удобной количественной характеристикой вектора является норма вектора. В вычислительной математике используются следующие три нормы (3)

$$¿\vee x\vee¿_1=\sum_{i=1}^{n}¿x_i\vee,\ ¿\vee x\vee¿_2=¿¿\ ¿\vee x\vee¿_\infty=max_{1\le i\le n}|x_i|. \qquad (3)$$

За норму матрицы принимают максимальную величину, на которую преобразование, описываемое матрицей, может растянуть любой ненулевой вектор в выбранной норме $\|A\|=max_{x\neq 0}\frac{¿\vee Ax\vee ¿}{¿|x|\vee ¿¿}$. Векторным нормам подчинены следующие нормы матрицы (4)

$$¿\vee A\vee ¿_1=max_{1\leq j\leq n}\sum_{i=1}^{n}|a_{ij}|,$$

$$¿\vee A\vee ¿_2=max_{1\leq j\leq n}\sqrt{\lambda_i(A^T A)},$$

$$¿\vee A\vee ¿_\infty=max_{1\leq i\leq n}\sum_{j=1}^{n}|a_{ij}|,$$

где λ_i – собственные числа матрицы A. Задача вычисления вектора X может быть плохо или хорошо обусловлена.

Обусловленность задачи решения систем линейных алгебраических уравнений

Рассмотрим случай, когда элементы матрицы A заданы точно, а вектор-столбец свободных членов – приближенно. Оценки для абсолютной и относительной погрешности (5)

$$\Delta(X^{¿})\leq v_\Delta \Delta(B^{¿}),$$

$$\delta(X^{¿})\leq v_\delta \delta(B^{¿}),$$

где $v_\Delta=¿\vee A^{-1}\vee ¿$- абсолютное число обусловленности, а $v_\delta=¿\vee A^{-1}\vee ¿\frac{¿\vee B\vee ¿}{¿\vee X\vee ¿}$ - относительное число обусловленности (естественное число обусловленности). Максимальное естественное число обусловленности (6)

$$max_{x\neq 0}v_\delta(X)=max_{x\neq 0}¿\vee A^{-1}\vee ¿\frac{¿\vee AX\vee ¿}{¿\vee X\vee ¿}=¿\vee A^{-1}\vee ¿\cdot\vee ¿A\vee ¿=cond(A) \quad (6)$$

называют стандартным числом обусловленности.

$$\delta(X^{¿})\leq cond(A)\delta(B^{¿}).$$

Если элементы матрицы A заданы приближенно и равны $A^{¿}$, а вектор-столбец свободных членов – точно, тогда оценка относительной погрешности (7)

$$\delta(X^{¿})\leq cond(A)\delta(A^{¿}), \quad (7)$$

где $\delta(X^{¿})=\frac{\|X-X^{¿}\|}{\|X\|}$ и $\delta(A^{¿})=\frac{¿\vee A-A^{¿}\vee ¿}{¿\vee A\vee ¿}$.

Если с погрешностью заданы как коэффициенты матрицы, так и элементы вектора свободных членов, то справедливо неравенство (8)

$$\delta(X^{¿})\leq cond(A)\left(\delta(A^{¿})+\delta(B^{¿})\right). \quad (8)$$

Использование wxMaxima для подсчета обратной матрицы

Матрица A – невырожденная, следовательно существует единственная обратная матрица A^{-1}. Для ее подсчета используется свободно распространяемый пакет системы компьютерной алгебры wxMaxima. Входная матрица задаётся с помощью выражения **matrix**(*стр1, стр2, ... стрN*), а обратная получается с помощью функции **invert**(*M*) (рисунок 1)

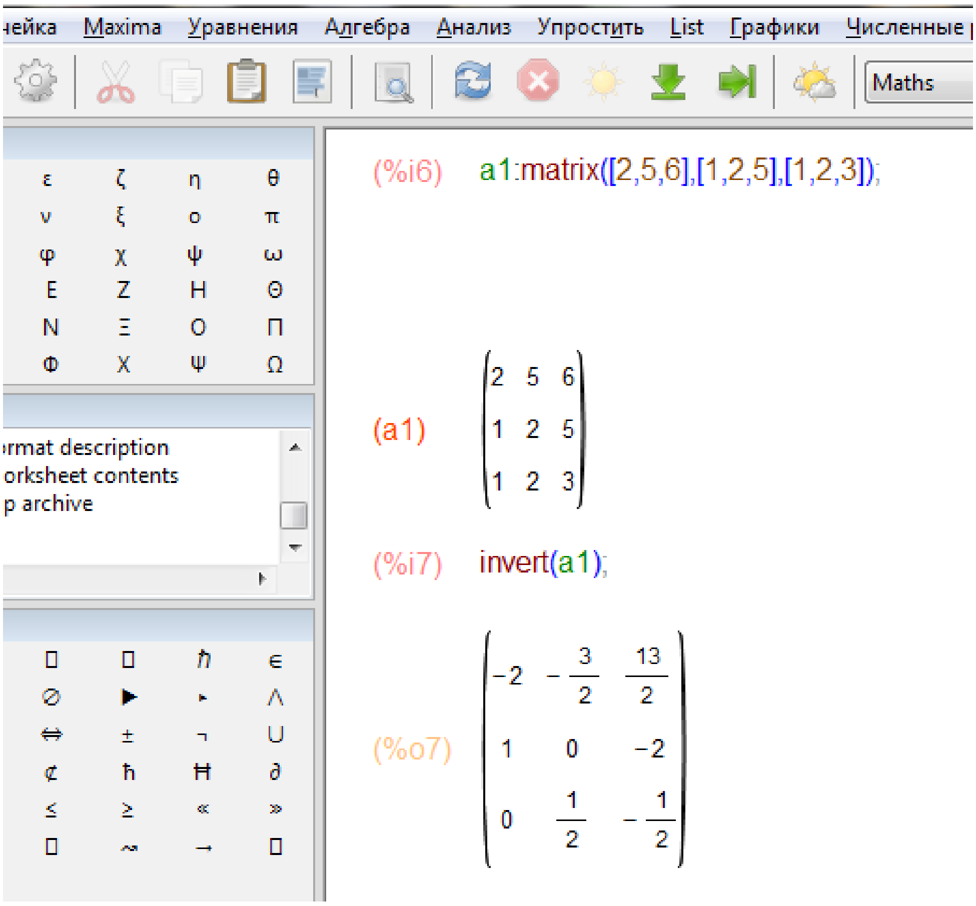


Рисунок 1 – Вычисление обратной матрицы с помощью функции *invert*

Порядок выполнения работы.

1 Составить подпрограмму для решения системы линейных уравнений методом Гаусса и методом обратной матрицы.

2 Решить систему, подсчитать стандартное число обусловленности, используя тип данных с двойной точностью. Подсчет обратной матрицы производить с помощью системы компьютерной алгебры wxMaxima.

3 С помощью встроенной функции – генератора случайных чисел, добавить ошибки в вектор свободных членов. Найти решение новой

системы, стандартное число обусловленности (6) и оценку стандартного числа обусловленности (7).

4 С помощью встроенной функции – генератора случайных чисел, добавить ошибки в значения элементов матрицы. Найти решение новой системы, стандартное число обусловленности и оценку стандартного числа обусловленности.

5 Добавить ошибки в значения элементов матрицы и вектора свободных членов. Найти решение новой системы, стандартное число обусловленности и оценку стандартного числа обусловленности.

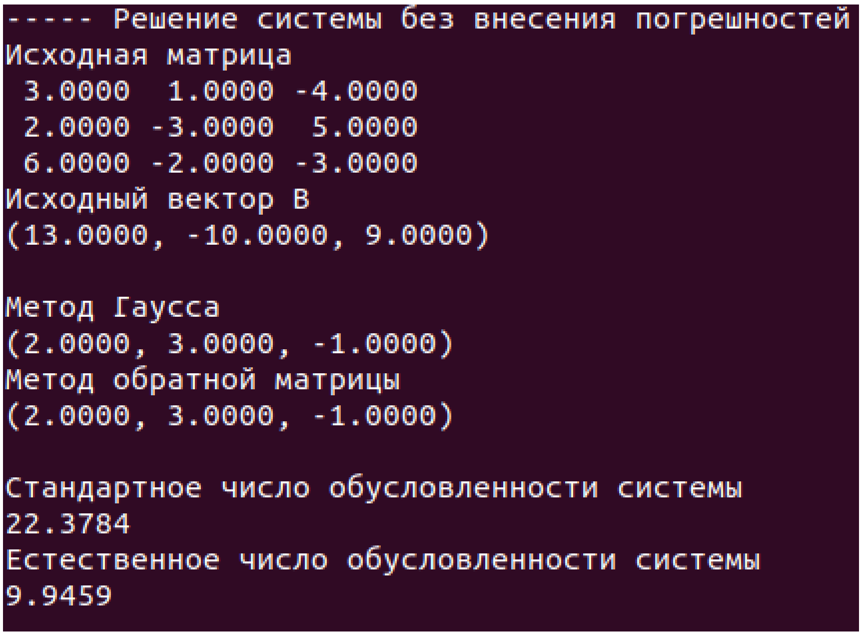
6 Сделать выводы по полученным значениям.

Выполнение работы.

Матрица: $\left(\begin{array}{ccc|c} 3 & 1 & -4 & 13 \\ 2 & -3 & 5 & -10 \\ 6 & -2 & -3 & 9 \end{array}\right)$

Исследование с исходной матрицей.

1. При помощи метода Гаусса и метода обратной матрицы было найдено решение матрицы. Была найдена обратная матрица с помощью метода Гаусса. Результаты представленны на рис. 2.



```
----- Решение системы без внесения погрешностей
Исходная матрица
 3.0000  1.0000 -4.0000
 2.0000 -3.0000  5.0000
 6.0000 -2.0000 -3.0000
Исходный вектор B
(13.0000, -10.0000, 9.0000)

Метод Гаусса
(2.0000, 3.0000, -1.0000)
Метод обратной матрицы
(2.0000, 3.0000, -1.0000)

Стандартное число обусловленности системы
22.3784
Естественное число обусловленности системы
9.9459
```

Рисунок 2 — Решение матрицы методами Гаусса и обратной матрицы.

Видно, что результаты для обоих методов совпадают и равны

X = $\begin{pmatrix} 2 \\ 3 \\ -1 \end{pmatrix}$. Стандартное число обусловленности оказалось равным $cond\,A = \|A^{-1}\| * \|A\| = 22{,}3784$ Было найдено естественное число обусловленности по формуле $\nu_\delta = \|A^{-1}\| \frac{\|B\|}{\|X\|}$ равное $\nu_\delta = 9{,}9459$.

2. В каждый член вектора B была внесена случайная погрешность, лежащая в промежутке от -0,1 до 0,1, после чего рассчеты были повторены. Результаты представленны на рис. 3.

```
----- Решение системы с внесением погрешностей в вектор B
Исходная матрица
 3.0000  1.0000 -4.0000
 2.0000 -3.0000  5.0000
 6.0000 -2.0000 -3.0000
Возумущенный вектор B
(12.9624, -9.9740, 9.0908)

Вектор-решение методом Гаусса
(1.9712, 2.9175, -1.0328)
Вектор-решение методом обратной матрицы
(1.9712, 2.9175, -1.0328)

Стандартное число обусловленности системы
22.3784
Оценка стандартного числа обусловленности системы
4.9752
Естественное число обусловленности системы
10.0777
```

Рисунок 3 - Решение матрицы с погрешностями в векторе B. По рисунку видно, что из-за внесения погрешностей изменился результат, который стал равен $X^* = \begin{pmatrix} 1{,}9712 \\ 2{,}9175 \\ -1{,}0328 \end{pmatrix}$, при этом результат полученный разными методами совпадает, как и до внесения погрешностей. Абсолютная погрешность будет равна ΔX = 0,0724, относительная погрешность δX = 0,0193, δB = 0,0039. Стандартное число обусловленности оказалась равной $cond\,A = \|A^{-1}\| * \|A\| = 22{,}3784$ Было найдено естественное число обусловленности по формуле

$\nu_\delta = \|A^{-1}\| \frac{\|B\|}{\|X\|}$ равное $\nu_\delta = 10{,}0777$. Оценка стандартного числа обусловленности - $\nu_\delta = \frac{(x^*)}{(B^*)}$ = 4,9752.

3. Были внесены случайные погрешности в каждый член исходной матрицы, величина погрешности лежит между 0,1 и 0,1, после чего рассчеты были повторены. Результаты представленны на рис. 4.

```
----- Решение системы с внесением погрешностей в матрицу системы
Возумущенная матрица системы
 2.9116  0.9232 -3.9032
 2.0410 -2.9088  5.0214
 6.0686 -2.0258 -3.0278
Исходный вектор B
(13.0000, -10.0000, 9.0000)

Вектор-решение методом Гаусса
(2.1810, 3.4261, -0.8933)
Вектор-решение методом обратной матрицы
(2.1810, 3.4261, -0.8933)

Стандартное число обусловленности системы
22.7459
Оценка стандартного числа обусловленности системы
7.2096
Естественное число обусловленности системы
9.3682
```

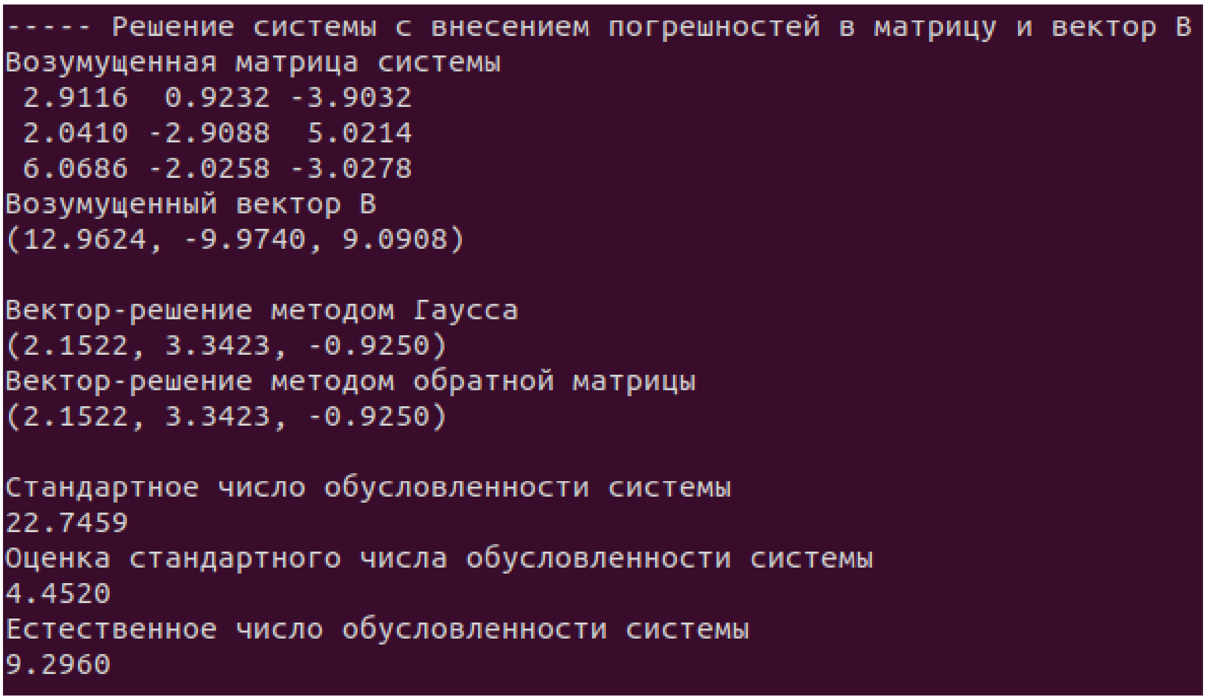
Рисунок 4 — Результаты решения матрицы с внесенными погрешностями в каждый её член.

По рисунку видно, что из-за внесения погрешностей изменился результат вычисления, но при этом результаты найденные разными способами одинаковы. Полученный результат $X^* = \begin{pmatrix} 2{,}181 \\ 3{,}4261 \\ -0{,}8933 \end{pmatrix}$.

Абсолютная погрешность будет равна ΔX = 0,4168, относительная погрешность δX = 0,1114, δA = 0,0038. Стандартное число обусловленности оказалась равной $cond\, A = \|A^{-1}\| * \|A\| = 22{,}7459$ Было найдено естественное число обусловленности по формуле

$\nu_\delta = \|A^{-1}\| \frac{\|B\|}{\|X\|}$ равное $\nu_\delta = 9{,}3682$. Оценка стандартного числа обусловленности - $\nu_\delta = \frac{(x^*)}{(A^*)}$ = 7,2096.

4. Была внесена случайная погрешность, лежащая в промежутке от -0,1 до 0,1, в каждый член матрицы A и в каждый член вектора B одновременно, после чего были повторены рассчеты. Результаты представленны на рис. 5.



```
----- Решение системы с внесением погрешностей в матрицу и вектор B
Возумущенная матрица системы
 2.9116  0.9232 -3.9032
 2.0410 -2.9088  5.0214
 6.0686 -2.0258 -3.0278
Возумущенный вектор B
(12.9624, -9.9740, 9.0908)

Вектор-решение методом Гаусса
(2.1522, 3.3423, -0.9250)
Вектор-решение методом обратной матрицы
(2.1522, 3.3423, -0.9250)

Стандартное число обусловленности системы
22.7459
Оценка стандартного числа обусловленности системы
4.4520
Естественное число обусловленности системы
9.2960
```

Рисунок 5 — Рассчеты с внессением погрешностей каждый член в матрицы A и вектора B одновременно.

По рисунку видно, что из-за внесения погрешностей изменился результат, который стал равен $X^* = \begin{pmatrix} 2{,}1522 \\ 3{,}3423 \\ -0{,}9250 \end{pmatrix}$. Абсолютная погрешность будет равна ΔX = 0,3398, относительная погрешность δX = 0,0908, δA = 0,0038, δB = 0,0039. Стандартное число обусловленности оказалась равной $cond\,A = \|A^{-1}\| * \|A\| = 22{,}7459$. Было найдено естественное число обусловленности по формуле

$\nu_\delta = \|A^{-1}\| \frac{\|B\|}{\|X\|}$ равное $\nu_\delta = 9{,}296$. Оценка стандартного числа обусловленности - $\nu_\delta = \frac{(x^*)}{(B^*)}$ = 4,452.

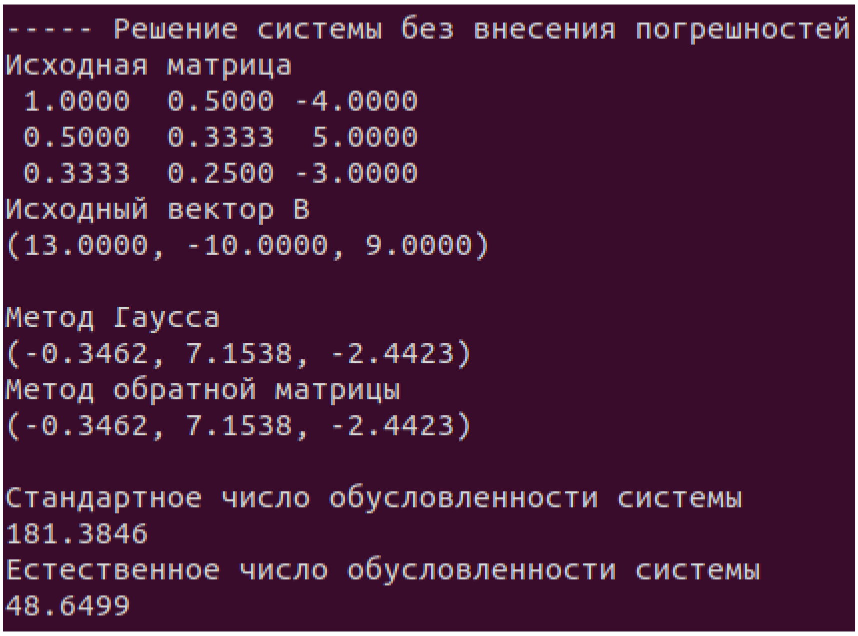
Вывод:

При внесении погрешности в вектор В, оценка стандартного числа обусловленности ν_δ = 4,9752. При внесении погрешностей в матрицу А, оценка стандартного числа обусловленности ν_δ = 7,2096. При внисении погрешностей одновременно в матрицу А и в вектор В, оценка стандартного числа обусловленности ν_δ = 4,452. Следовательно, максимальная оценка стандартного числа обусловленности при внесении погрешностей в матрциу А, минимальная при внесении погрешностей в матрицу А и вектор В одновременно, оценка стандартного числа обусловленности при внесении погрешностей только в вектор В лежит между максимальным и минимальным значениями.

Исследование с матрцией, с 1 и 2 столбцами из матрицы Гильберта.

$$\left(\begin{array}{ccc|c} 1 & 1/2 & -4 & 13 \\ 1/2 & 1/3 & 5 & -10 \\ 1/3 & 1/4 & -3 & 9 \end{array}\right)$$

1. При помощи метода Гаусса и метода обратной матрицы было найдено решение матрицы. Была найдена обратная матрица с помощью метода Гаусса. Результаты представленны на рис. 6.

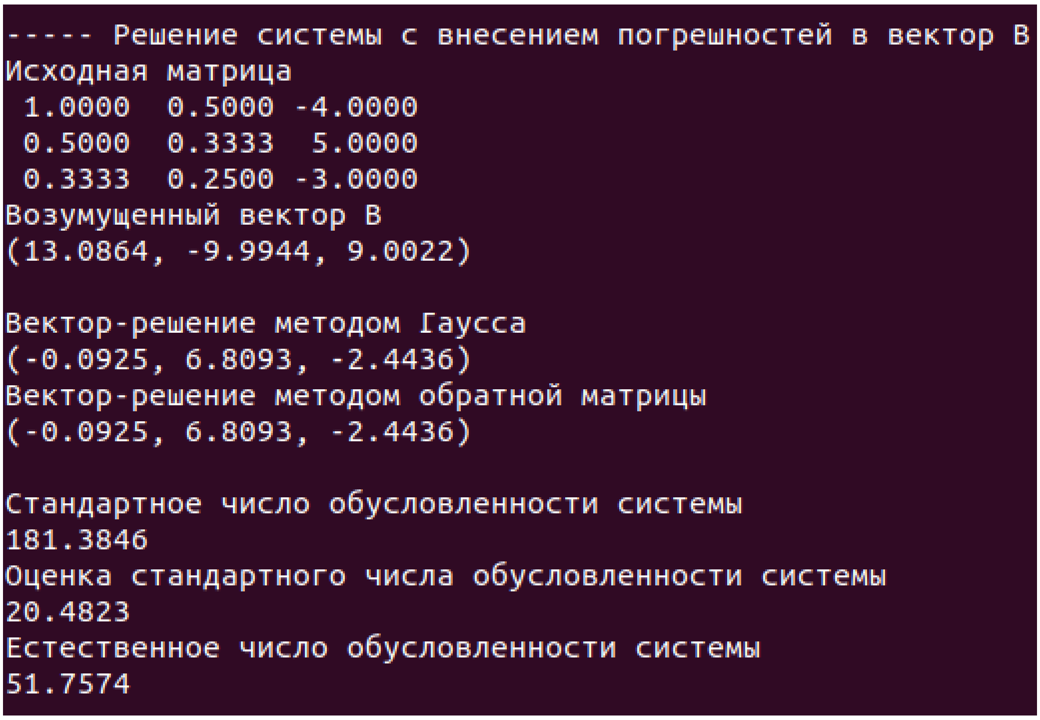


```
----- Решение системы без внесения погрешностей
Исходная матрица
 1.0000  0.5000 -4.0000
 0.5000  0.3333  5.0000
 0.3333  0.2500 -3.0000
Исходный вектор B
(13.0000, -10.0000, 9.0000)

Метод Гаусса
(-0.3462, 7.1538, -2.4423)
Метод обратной матрицы
(-0.3462, 7.1538, -2.4423)

Стандартное число обусловленности системы
181.3846
Естественное число обусловленности системы
48.6499
```

Рисунок 6 — Решение матрицы методами Гаусса и обратной матрицы. Видно, что результаты для обоих методов совпадают и равны

X = $\begin{pmatrix} -0{,}3462 \\ 7{,}1538 \\ -2{,}4423 \end{pmatrix}$. Стандартное число обусловленности оказалась равной $cond\,A = \|A^{-1}\| * \|A\| = 181{,}3846$ Было найдено естественное число обусловленности по формуле $\nu_\delta = \|A^{-1}\| \frac{\|B\|}{\|X\|}$ равное $\nu_\delta = 48{,}6499$.

2. В каждый член вектора В была внесена случайная погрешность, лежащая в промежутке от -0,1 до 0,1, после чего рассчеты были повторены. Результаты представленны на рис. 7.



```
----- Решение системы с внесением погрешностей в вектор B
Исходная матрица
 1.0000  0.5000 -4.0000
 0.5000  0.3333  5.0000
 0.3333  0.2500 -3.0000
Возумущенный вектор B
(13.0864, -9.9944, 9.0022)

Вектор-решение методом Гаусса
(-0.0925, 6.8093, -2.4436)
Вектор-решение методом обратной матрицы
(-0.0925, 6.8093, -2.4436)

Стандартное число обусловленности системы
181.3846
Оценка стандартного числа обусловленности системы
20.4823
Естественное число обусловленности системы
51.7574
```

Рисунок 7 - Решение матрицы с погрешностями в векторе В.

По рисунку видно, что из-за внесения погрешностей изменился результат, который стал равен $X^{*}=\begin{pmatrix}-0{,}0925\\ 6{,}8093\\ -2{,}4436\end{pmatrix}$, при этом результат полученный разными методами совпадает, как и до внесения погрешностей. Абсолютная погрешность будет равна ΔX = 0,332, относительная погрешность δX = 0,0439, δB = 0,0032. Стандартное число обусловленности оказалась равной $cond\, A = \|A^{-1}\| * \|A\| = 181{,}3846$. Было найдено естественное число обусловленности по формуле $\nu_{\delta} = \|A^{-1}\| \frac{\|B\|}{\|X\|}$ равное $\nu_{\delta} = 51{,}7574$.

Оценка стандартного числа обусловленности - $\nu_{\delta} = \frac{(x^{*})}{(B^{*})}$ = 20,4823.

3. Были внесены случайные погрешности в каждый член исходной матрицы, величина погрешности лежит между 0,1 и 0,1, после чего рассчеты были повторены. Результаты представленны на рис. 8.

```
----- Решение системы с внесением погрешностей в матрицу системы
Возумущенная матрица системы
 0.9830  0.5240 -4.0348
 0.4298  0.3309  4.9626
 0.3179  0.2746 -2.9872
Исходный вектор B
(13.0000, -10.0000, 9.0000)

Вектор-решение методом Гаусса
(-0.3251, 6.6867, -2.4328)
Вектор-решение методом обратной матрицы
(-0.3251, 6.6867, -2.4328)

Стандартное число обусловленности системы
147.7740
Оценка стандартного числа обусловленности системы
5.8529
Естественное число обусловленности системы
41.7773
```

Рисунок 8 — Результаты решения матрицы с внесенными погрешностями в каждый её член.

По рисунку видно, что из-за внесения погрешностей изменился результат вычисления, но при этом результаты найденные разными способами одинаковы. Полученный результат $X^{*}=\begin{pmatrix}-0{,}3251\\6{,}6864\\-2{,}4328\end{pmatrix}$.

Абсолютная погрешность будет равна ΔX = 0,4444, относительная погрешность δX = 0,0587, δA = 0,0017. Стандартное число обусловленности оказалась равной $cond\,A=\|A^{-1}\|*\|A\|=147{,}774$. Было найдено естественное число обусловленности по формуле $\nu_{\delta}=\|A^{-1}\|\frac{\|B\|}{\|X\|}$ равное $\nu_{\delta}=41{,}7773$. Оценка стандартного числа обусловленности - $\nu_{\delta}=\frac{(x^{*})}{(A^{*})}$ = 5,8529.

4. Была внесена случайная погрешность, лежащая в промежутке от -0,1 до 0,1, в каждый член матрицы A и в каждый член вектора B одновременно, после чего были повторены рассчеты. Результаты представленны на рис. 9.

```
----- Решение системы с внесением погрешностей в матрицу и вектор В
Возумущенная матрица системы
 0.9830  0.5240 -4.0348
 0.4298  0.3309  4.9626
 0.3179  0.2746 -2.9872
Возумущенный вектор В
(13.0864, -9.9944, 9.0022)

Вектор-решение методом Гаусса
(-0.1029, 6.4280, -2.4336)
Вектор-решение методом обратной матрицы
(-0.1029, 6.4280, -2.4336)

Стандартное число обусловленности системы
147.7740
Оценка стандартного числа обусловленности системы
8.5546
Естественное число обусловленности системы
53.9564
```

Рисунок 9 — Рассчеты с внессением погрешностей каждый член в матрицы A и вектора B одновременно.

По рисунку видно, что из-за внесения погрешностей изменился результат, который стал равен $X^{*} = \begin{pmatrix} -0,1029 \\ 6,428 \\ -2,4336 \end{pmatrix}$. Абсолютная погрешность будет равна ΔX = 0,6934, относительная погрешность δX = 0,0916, δA = 0,0017, δB = 0,0032. Стандартное число обусловленности оказалась равной $cond\, A = \|A^{-1}\| * \|A\| = 147{,}774$. Было найдено естественное число обусловленности по формуле $\nu_{\delta} = \|A^{-1}\| \frac{\|B\|}{\|X\|}$ равное $\nu_{\delta} = 53{,}9564$. Оценка стандартного числа обусловленности - $\nu_{\delta} = \frac{(x^{*})}{(B^{*})}$ = 8,5546.

Вывод:

При внесении погрешности в вектор B, оценка стандартного числа обусловленности ν_δ = 20,4823. При внесении погрешностей в матрицу A, оценка стандартного числа обусловленности ν_δ = 5,8529. При внисении погрешностей одновременно в матрицу A и в вектор B, оценка стандартного числа обусловленности ν_δ = 8,5546. Следовательно, при внесении погрешности в вектор B получается максимальная оценка стандартного числа обусловленности, минимальная оценка получается при внисении погрешности в матрицу A, оценка стандартного числа обусловленности при внисении погрешностей в матрицу A и в вектор B одновременно находится между максимальным и минимальным значениями.

Разработанный программный код см. в приложении А.

Выводы.

Была изучена стандартная обусловленность задач решения СЛУ при различных погрешностях входных данных.

При внесении погрешностей исключительно в матрицу, оценка стандартного числа обусловленности получается максимальной, при внесении погрешностей в вектор получается или в вектор и матрицу одновременно оценка стандартного числа обусловленности получается меньше, при этом минимальной может быть любая из двух.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.cpp

```
#include <iostream>
#include <vector>
#include <cmath>
#include <ctime>
#include <cstdlib>
#include <iomanip>

void calculate1(const std::vector<double>& matrix, const::std::vector<double>& vectorB);
void calculate2(const std::vector<double>& matrix, const::std::vector<double>& vectorB);
void printMatrix(const std::vector<double>& m);
void printVector(const std::vector<double>& v);
std::vector<double> gaussianMethod(std::vector<double> m, std::vector<double> b);
std::vector<double> inverseMatrixMethod(std::vector<double> inv, std::vector<double> b);
std::vector<double> getInverseMatrix(std::vector<double> m);
double vectorNorm(const std::vector<double>& v);
double matrixNorm(const std::vector<double>& m);
std::vector<double> setInaccuracyInVector1(const std::vector<double>& v);
std::vector<double> setInaccuracyInMatrix1(const std::vector<double>& m);
std::vector<double> setInaccuracyInVector2(const std::vector<double>& v);
std::vector<double> setInaccuracyInMatrix2(const std::vector<double>& m);
double cond(const std::vector<double>& m, const std::vector<double>& inv);
double relativeInaccuracyVector(const std::vector<double>& v, const std::vector<double>& in);
double relativeInaccuracyMatrix(const std::vector<double>& m, const std::vector<double>& in);
const int matrix_side = 3;
const std::vector<double> __matrix = {3,  1, -4,
                                      2, -3,  5,
                                      6, -2, -3
};
const std::vector<double> __gilbertMatrix = {1, 1 / 2.0, -4,
                                             1 / 2.0, 1 / 3.0, 5,
                                             1 / 3.0, 1 / 4.0, -3
};
const std::vector<double> __vectorB = {13, -10, 9};
const std::vector<double> __inverseMatrix = {19/37.0, 11/37.0, -7/37.0,
                                             36/37.0, 15/37.0, -23/37.0,
```

```
                                                               14/37.0,
12/37.0, -11/37.0
    };

    int main(){
        calculate1(__matrix, __vectorB);
        calculate2(__gilbertMatrix, __vectorB);
        return 0;
    }

    void calculate1(const std::vector<double>& matrix, const::std::vector<double>& vectorB){
        std::vector<double> solutionG = gaussianMethod(matrix, vectorB);
        std::vector<double> solutionI = inverseMatrixMethod(getInverseMatrix(matrix), vectorB);
        std::vector<double> in_matrix = setInaccuracyInMatrix1(matrix);
        std::vector<double> in_vectorB = setInaccuracyInVector1(vectorB);
        std::vector<double> in1_solutionG = gaussianMethod(in_matrix, vectorB);
        std::vector<double> in1_solutionI = inverseMatrixMethod(getInverseMatrix(in_matrix), vectorB);
        std::vector<double> in2_solutionG = gaussianMethod(matrix, in_vectorB);
        std::vector<double> in2_solutionI = inverseMatrixMethod(getInverseMatrix(matrix), in_vectorB);
        std::vector<double> in3_solutionG = gaussianMethod(in_matrix, in_vectorB);
        std::vector<double> in3_solutionI = inverseMatrixMethod(getInverseMatrix(in_matrix), in_vectorB);
        std::cout << std::fixed;
        std::cout << std::setprecision(4);
        std::cout << "----- Решение системы без внесения погрешностей -----" << std::endl;
        std::cout << "Исходная матрица" << std::endl;
        printMatrix(matrix);
        std::cout << "Исходный вектор B" << std::endl;
        printVector(vectorB);
        std::cout << std::endl;
        std::cout << "Метод Гаусса" << std::endl;
        printVector(solutionG);
        std::cout << "Метод обратной матрицы" << std::endl;
        printVector(solutionI);
        std::cout << std::endl;
        std::cout << "Стандартное число обусловленности системы" << std::endl;
        std::cout << cond(matrix, getInverseMatrix(matrix)) << std::endl;
        std::cout << "Естественное число обусловленности системы" << std::endl;
        std::cout << matrixNorm(getInverseMatrix(matrix)) * vectorNorm(vectorB) / vectorNorm(solutionG) << std::endl;
        std::cout << std::endl;
```

```
        std::cout << "----- Решение системы с внесением погрешностей в матрицу системы -----" << std::endl;
        std::cout << "Возумущенная матрица системы" << std::endl;
        printMatrix(in_matrix);
        std::cout << "Исходный вектор B" << std::endl;
        printVector(vectorB);
        std::cout << std::endl;
        std::cout << "Вектор-решение методом Гаусса" << std::endl;
        printVector(in1_solutionG);
        std::cout << "Вектор-решение методом обратной матрицы" << std::endl;
        printVector(in1_solutionI);
        std::cout << std::endl;
        std::cout << "Стандартное число обусловленности системы" << std::endl;
        std::cout << cond(in_matrix, getInverseMatrix(in_matrix)) << std::endl;
        std::cout << "Оценка стандартного числа обусловленности системы" << std::endl;
        std::cout << relativeInaccuracyVector(solutionG, in1_solutionG) / relativeInaccuracyMatrix(matrix, in_matrix) << std::endl;
        std::cout << "Естественное число обусловленности системы" << std::endl;
        std::cout << matrixNorm(getInverseMatrix(in_matrix)) * vectorNorm(vectorB) / vectorNorm(in1_solutionG) << std::endl;
        std::cout << std::endl;
        std::cout << "----- Решение системы с внесением погрешностей в вектор B -----" << std::endl;
        std::cout << "Исходная матрица" << std::endl;
        printMatrix(matrix);
        std::cout << "Возумущенный вектор B" << std::endl;
        printVector(in_vectorB);
        std::cout << std::endl;
        std::cout << "Вектор-решение методом Гаусса" << std::endl;
        printVector(in2_solutionG);
        std::cout << "Вектор-решение методом обратной матрицы" << std::endl;
        printVector(in2_solutionI);
        std::cout << std::endl;
        std::cout << "Стандартное число обусловленности системы" << std::endl;
        std::cout << cond(matrix, getInverseMatrix(matrix)) << std::endl;
        std::cout << "Оценка стандартного числа обусловленности системы" << std::endl;
        std::cout << relativeInaccuracyVector(solutionG, in2_solutionG) / relativeInaccuracyVector(vectorB, in_vectorB) << std::endl;
        std::cout << "Естественное число обусловленности системы" << std::endl;
        std::cout << matrixNorm(getInverseMatrix(matrix)) * vectorNorm(vectorB) / vectorNorm(in2_solutionG) << std::endl;
        std::cout << std::endl;
        std::cout << "----- Решение системы с внесением погрешностей в матрицу и вектор B -----" << std::endl;
```

```
            std::cout << "Возумущенная матрица системы" << std::endl;
            printMatrix(in_matrix);
            std::cout << "Возумущенный вектор B" << std::endl;
            printVector(in_vectorB);
            std::cout << std::endl;
            std::cout << "Вектор-решение методом Гаусса" << std::endl;
            printVector(in3_solutionG);
            std::cout << "Вектор-решение методом обратной матрицы" <<
std::endl;
            printVector(in3_solutionI);
            std::cout << std::endl;
            std::cout << "Стандартное число обусловленности системы" <<
std::endl;
            std::cout << cond(in_matrix, getInverseMatrix(in_matrix)) <<
std::endl;
            std::cout << "Оценка стандартного числа обусловленности
системы" << std::endl;
            std::cout << relativeInaccuracyVector(solutionG,
in3_solutionG) / (relativeInaccuracyVector(vectorB, in_vectorB) +
            relativeInaccuracyMatrix(matrix, in_matrix)) << std::endl;
            std::cout << "Естественное число обусловленности системы" <<
std::endl;
            std::cout << matrixNorm(getInverseMatrix(matrix)) *
            vectorNorm(vectorB) / vectorNorm(in3_solutionG) <<
std::endl;
            std::cout << std::endl;
      }

      void calculate2(const std::vector<double>& matrix,
const::std::vector<double>& vectorB){
            std::vector<double> solutionG = gaussianMethod(matrix,
vectorB);
            std::vector<double> solutionI =
inverseMatrixMethod(getInverseMatrix(matrix), vectorB);
            std::vector<double> in_matrix =
setInaccuracyInMatrix2(matrix);
            std::vector<double> in_vectorB =
setInaccuracyInVector2(vectorB);
            std::vector<double> in1_solutionG =
gaussianMethod(in_matrix, vectorB);
            std::vector<double> in1_solutionI =
inverseMatrixMethod(getInverseMatrix(in_matrix), vectorB);
            std::vector<double> in2_solutionG = gaussianMethod(matrix,
in_vectorB);
            std::vector<double> in2_solutionI =
inverseMatrixMethod(getInverseMatrix(matrix), in_vectorB);
            std::vector<double> in3_solutionG =
gaussianMethod(in_matrix, in_vectorB);
            std::vector<double> in3_solutionI =
inverseMatrixMethod(getInverseMatrix(in_matrix), in_vectorB);
            std::cout << std::fixed;
            std::cout << std::setprecision(4);
            std::cout << "----- Решение системы без внесения
погрешностей -----" << std::endl;
            std::cout << "Исходная матрица" << std::endl;
            printMatrix(matrix);
```

```
        std::cout << "Исходный вектор B" << std::endl;
        printVector(vectorB);
        std::cout << std::endl;
        std::cout << "Метод Гаусса" << std::endl;
        printVector(solutionG);
        std::cout << "Метод обратной матрицы" << std::endl;
        printVector(solutionI);
        std::cout << std::endl;
        std::cout << "Стандартное число обусловленности системы" << std::endl;
        std::cout << cond(matrix, getInverseMatrix(matrix)) << std::endl;
        std::cout << "Естественное число обусловленности системы" << std::endl;
        std::cout << matrixNorm(getInverseMatrix(matrix)) * vectorNorm(vectorB) / vectorNorm(solutionG) << std::endl;
        std::cout << std::endl;
        std::cout << "----- Решение системы с внесением погрешностей в матрицу системы -----" << std::endl;
        std::cout << "Возумущенная матрица системы" << std::endl;
        printMatrix(in_matrix);
        std::cout << "Исходный вектор B" << std::endl;
        printVector(vectorB);
        std::cout << std::endl;
        std::cout << "Вектор-решение методом Гаусса" << std::endl;
        printVector(in1_solutionG);
        std::cout << "Вектор-решение методом обратной матрицы" << std::endl;
        printVector(in1_solutionI);
        std::cout << std::endl;
        std::cout << "Стандартное число обусловленности системы" << std::endl;
        std::cout << cond(in_matrix, getInverseMatrix(in_matrix)) << std::endl;
        std::cout << "Оценка стандартного числа обусловленности системы" << std::endl;
        std::cout << relativeInaccuracyVector(solutionG, in1_solutionG) / relativeInaccuracyMatrix(matrix, in_matrix) << std::endl;
        std::cout << "Естественное число обусловленности системы" << std::endl;
        std::cout << matrixNorm(getInverseMatrix(in_matrix)) * vectorNorm(vectorB) / vectorNorm(in1_solutionG) << std::endl;
        std::cout << std::endl;
        std::cout << "----- Решение системы с внесением погрешностей в вектор B -----" << std::endl;
        std::cout << "Исходная матрица" << std::endl;
        printMatrix(matrix);
        std::cout << "Возумущенный вектор B" << std::endl;
        printVector(in_vectorB);
        std::cout << std::endl;
        std::cout << "Вектор-решение методом Гаусса" << std::endl;
        printVector(in2_solutionG);
        std::cout << "Вектор-решение методом обратной матрицы" << std::endl;
        printVector(in2_solutionI);
```

```
        std::cout << std::endl;
        std::cout << "Стандартное число обусловленности системы" <<
std::endl;
        std::cout << cond(matrix, getInverseMatrix(matrix)) <<
std::endl;
        std::cout << "Оценка стандартного числа обусловленности
системы" << std::endl;
        std::cout << relativeInaccuracyVector(solutionG,
in2_solutionG) / relativeInaccuracyVector(vectorB, in_vectorB) <<
std::endl;
        std::cout << "Естественное число обусловленности системы" <<
std::endl;
        std::cout << matrixNorm(getInverseMatrix(matrix)) *
vectorNorm(vectorB) / vectorNorm(in2_solutionG) << std::endl;
        std::cout << std::endl;
        std::cout << "----- Решение системы с внесением погрешностей
в матрицу и вектор B -----" << std::endl;
        std::cout << "Возумущенная матрица системы" << std::endl;
        printMatrix(in_matrix);
        std::cout << "Возумущенный вектор B" << std::endl;
        printVector(in_vectorB);
        std::cout << std::endl;
        std::cout << "Вектор-решение методом Гаусса" << std::endl;
        printVector(in3_solutionG);
        std::cout << "Вектор-решение методом обратной матрицы" <<
std::endl;
        printVector(in3_solutionI);
        std::cout << std::endl;
        std::cout << "Стандартное число обусловленности системы" <<
std::endl;
        std::cout << cond(in_matrix, getInverseMatrix(in_matrix)) <<
std::endl;
        std::cout << "Оценка стандартного числа обусловленности
системы" << std::endl;
        std::cout << relativeInaccuracyVector(solutionG,
in3_solutionG) / (relativeInaccuracyVector(vectorB, in_vectorB) +
        relativeInaccuracyMatrix(matrix, in_matrix)) << std::endl;
        std::cout << "Естественное число обусловленности системы" <<
std::endl;
        std::cout << matrixNorm(getInverseMatrix(matrix)) *
        vectorNorm(vectorB) / vectorNorm(in3_solutionG) <<
std::endl;
        std::cout << std::endl;
    }

    void printMatrix(const std::vector<double>& m){
        for (int i = 0; i < matrix_side; ++i){
            for (int j = 0; j < matrix_side; ++j){
                std::cout << std::setw(7);
                std::cout << m.at(i * matrix_side + j) << " ";
            }
            std::cout << std::endl;
        }
    }

    void printVector(const std::vector<double>& v){
```

```
        std::cout << "(";
        for (int i = 0; i < v.size() - 1; ++i)
            std::cout << v.at(i) << ", ";
        std::cout << v.at(v.size() - 1);
        std::cout << ")" << std::endl;
    }

    std::vector<double>     gaussianMethod(std::vector<double>     m,
std::vector<double> b){
        int i, j, k;
        double nu, prime;
        for (i = 0; i < matrix_side; ++i){
            prime = m.at(i * matrix_side + i);
            for (j = 0; j < matrix_side; ++j){
                if (j == i){
                    for (k = 0; k < matrix_side; ++k)
                        m.at(i * matrix_side + k) /= prime;
                    b.at(i) /= prime;
                    continue;
                }
                nu  =  -m.at(j  *  matrix_side  +  i)  /  m.at(i  *
matrix_side + i);
                for (k = 0; k < matrix_side; ++k)
                    m.at(j * matrix_side + k) += nu * m.at(i *
matrix_side + k);
                b.at(j) += nu * b.at(i);
            }
        }
        return b;
    }

    std::vector<double>  inverseMatrixMethod(std::vector<double>  inv,
std::vector<double> b){
        int i, j;
        std::vector<double> result;
        result.resize(matrix_side);
        for (i = 0; i < matrix_side; ++i){
            double element = 0;
            for (j = 0; j < matrix_side; ++j)
                element += inv.at(i * matrix_side + j) * b.at(j);
            result.at(i) = element;
        }
        return result;
    }

    double vectorNorm(const std::vector<double>& v){
        double norm = 0;
        for (int i = 0; i < matrix_side; ++i)
            norm += std::fabs(v.at(i));
        return norm;
    }

    double matrixNorm(const std::vector<double>& m){
        double norm = 0, _norm = 0;
        for (int j = 0; j < matrix_side; ++j){
            _norm = 0;
```

```
                for (int i = 0; i < matrix_side; ++i){
                    _norm += std::fabs(m.at(i * matrix_side + j));
                }
                norm = norm > _norm ? norm : _norm;
            }
            return norm;
        }

        std::vector<double>                    setInaccuracyInVector1(const
    std::vector<double>& v){
            std::srand(std::time(NULL) + reinterpret_cast<size_t>(&v));
            double inaccuracy;
            std::vector<double> new_vector;/*
            new_vector.reserve(v.size());
            for (const double& x: v){
                inaccuracy = (std::rand() % 1000 - 500) / 5000.0; //
    [-0.1; 0.1]
                new_vector.push_back(x + inaccuracy);
            }*/
            new_vector.push_back(12.9624);
            new_vector.push_back(-9.9740);
            new_vector.push_back(9.0908);
            return new_vector;
        }

        std::vector<double>                    setInaccuracyInMatrix1(const
    std::vector<double>& m){
            std::srand(std::time(NULL) + reinterpret_cast<size_t>(&m));
            double inaccuracy;
            std::vector<double> new_vector;/*
            new_vector.reserve(m.size());
            for (const double& x: m){
                inaccuracy = (std::rand() % 1000 - 500) / 5000.0; //
    [-0.1; 0.1]
                new_vector.push_back(x + inaccuracy);
            }*/
            new_vector.push_back(2.9116);
            new_vector.push_back(0.9232);
            new_vector.push_back(-3.9032);
            new_vector.push_back(2.0410);
            new_vector.push_back(-2.9088);
            new_vector.push_back(5.0214);
            new_vector.push_back(6.0686);
            new_vector.push_back(-2.0258);
            new_vector.push_back(-3.0278);
            return new_vector;
        }

        std::vector<double>                    setInaccuracyInVector2(const
    std::vector<double>& v){
            std::srand(std::time(NULL) + reinterpret_cast<size_t>(&v));
            double inaccuracy;
            std::vector<double> new_vector;
            new_vector.push_back(13.0864);
            new_vector.push_back(-9.9944);
            new_vector.push_back(9.0022);
```

```
        return new_vector;
    }

    std::vector<double>                 setInaccuracyInMatrix2(const
std::vector<double>& m){
        std::srand(std::time(NULL) + reinterpret_cast<size_t>(&m));
        double inaccuracy;
        std::vector<double> new_vector;
        new_vector.push_back(0.9830);
        new_vector.push_back(0.5240);
        new_vector.push_back(-4.0348);
        new_vector.push_back(0.4298);
        new_vector.push_back(0.3309);
        new_vector.push_back(4.9626);
        new_vector.push_back(0.3179);
        new_vector.push_back(0.2746);
        new_vector.push_back(-2.9872);
        return new_vector;
    }

    std::vector<double> getInverseMatrix(std::vector<double> m){
        std::vector<double> inverse;
        inverse.resize(matrix_side * matrix_side);
        for (int i = 0; i < matrix_side; ++i)
            inverse.at(i * matrix_side + i) = 1;
        int i, j, k;
        double nu, prime;
        for (i = 0; i < matrix_side; ++i){
            prime = m.at(i * matrix_side + i);
            for (j = 0; j < matrix_side; ++j){
                if (j == i){
                    for (k = 0; k < matrix_side; ++k){
                        m.at(i * matrix_side + k) /= prime;
                        inverse.at(i  *  matrix_side  +  k)  /=
prime;
                    }
                    continue;
                }
                nu  =  -m.at(j  *  matrix_side  +  i)  /  m.at(i  *
matrix_side + i);
                for (k = 0; k < matrix_side; ++k){
                    m.at(j * matrix_side + k) += nu * m.at(i *
matrix_side + k);
                    inverse.at(j  *  matrix_side  +  k)  +=  nu  *
inverse.at(i * matrix_side + k);
                }
            }
        }
        return inverse;
    }

    double     cond(const     std::vector<double>&     m,     const
std::vector<double>& inv){
        return matrixNorm(m) * matrixNorm(inv);
    }
```

```
    double   relativeInaccuracyVector(const   std::vector<double>&   v,
const std::vector<double>& in){
        std::vector<double> delta;
        delta.resize(v.size());
        for (int i = 0; i < v.size(); ++i)
            delta.at(i) = v.at(i) - in.at(i);
        return vectorNorm(delta) / vectorNorm(v);
    }

    double   relativeInaccuracyMatrix(const   std::vector<double>&   m,
const std::vector<double>& in){
        std::vector<double> delta;
        delta.resize(m.size());
        for (int i = 0; i < m.size(); ++i)
            delta.at(i) = m.at(i) - in.at(i);
        return matrixNorm(delta) / matrixNorm(m);
    }
```