Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Методы численного анализа

**ОТЧЁТ**

к лабораторной работе

на тему

Решение систем линейных алгебраических уравнений (СЛАУ)

методом Гаусса и с помощью его модификаций

Выполнил: студент группы 253503

Ярмак Вероника Сергеевна

Проверил: Анисимов Владимир Яковлевич

Минск 2022

СОДЕРЖАНИЕ

[1. Цель работы 2](https://docs.google.com/document/d/1PstHmKqzPsCO-CRAPCLlmcaynzw-Ain3yyu5brHn30k/edit#heading=h.30j0zll)

[2. Теоретические сведения 2](https://docs.google.com/document/d/1PstHmKqzPsCO-CRAPCLlmcaynzw-Ain3yyu5brHn30k/edit#heading=h.1fob9te)

[3. Решение тестовых примеров 5](https://docs.google.com/document/d/1PstHmKqzPsCO-CRAPCLlmcaynzw-Ain3yyu5brHn30k/edit#heading=h.26in1rg)

4. Блок схема для алгоритма решения СЛАУ методом Гаусса

[5. Выводы 8](https://docs.google.com/document/d/1PstHmKqzPsCO-CRAPCLlmcaynzw-Ain3yyu5brHn30k/edit#heading=h.lnxbz9)

# Цель работы

1. Изучить метод Гаусса и его модификации, составить алгоритм метода и программу его реализации, получить численное решение заданной СЛАУ;
2. Составить алгоритм решения СЛАУ указанными методами, применимый для организации вычислений на ЭВМ;
3. Составить программу решения СЛАУ по разработанному алгоритму;
4. Выполнить тестовые примеры и проверить правильность работы программы;

# Теоретические сведения

Задача отыскать решения СЛАУ с n неизвестными является одной из наиболее часто встречающихся вычислительных задач. Хотя задача решения СЛАУ сравнительно редко представляет самостоятельней интерес для приложений, от умения эффективно решать такие системы часто зависит сама возможность математического моделирования с применением ЭВМ разнообразных процессов. Значительная часть численных методов решения различных по своей природе задач (в особенности – нелинейных) включает в себя решение систем линейных уравнений как элементарный шаг соответствующего алгоритма.

СЛАУ обычно записывается в виде

j=1najxj=bi;i≤1≤n, или коротко Ax=b, где

 A = a11      a12   a1n a n1    a n2      a nm ;  a = x1 xn ;  b = b1 bn .

Здесь А и b заданы и требуется найти x.

**Метод Гаусса**

Одним из самых распространенных методов решения систем линейных уравнений является метод Гаусса. Этот метод (который также называют *методом последовательного исключения неизвестных*) известен в различных вариантах.

Вычисления с помощью метода Гаусса заключаются в последовательном исключении неизвестных из системы для преобразования ее к эквивалентной системе с верхней треугольной матрицей. Вычисления значений неизвестных производят на этапе обратного хода.

**1. Схема единственного деления.**

Рассмотрим сначала простейший вариант метода Гаусса.

Прямой ход состоит из *n* − 1 шагов исключения.

1-й шаг. Целью этого шага является исключение неизвестного *x*1 из уравнений с номерами *i* = 2, 3, …, *n*. Предположим, что коэффициент *a*11 ≠ 0. Будем называть его *главным элементом* 1-*го шага*.

Найдем величины

*qi*1*= ai*1/*a*11 (*i =*2, 3, …,*n*),

называемые *множителями*1-*го шага*. Вычтем последовательно из второго, третьего, …, *n-*го уравнений системы первое уравнение, умноженное соответственно на *q2*1*, q*31*, …, qn*1. Это позволит обратить в нуль коэффициенты при *x*1 во всех уравнениях, кроме первого. В результате получим эквивалентную систему

*a*11*x*1 + *a*12*x*2 + *a*13*x*3 + … + *a*1*nxn*= *b*1,

*a*22(1)*x*2 + *a*23(1)*x*3 + … + *a*2*n*(1)*xn*= *b*2(1),

*a*32(1)*x*2 + *a*33(1)*x*3 + … + *a*3*n*(1)*xn*= *b*3(1) ,

. . . . . . . . . . . . . . .

*an*2(1)*x*2 + *an*3(1)*x*3 + … + *ann*(1)*xn*= *bn*(1) .

в которой *aij*(1) и *bij*(1) вычисляются по формулам

*aij*(1) = *aij − qi*1*a*1*j*, *bi*(1) = *bi − qi*1*b*1.

2-й шаг. Целью этого шага является исключение неизвестного *x*2 из уравнений с номерами *i =*3, 4, …, *n*. Пусть *a*22(1) ≠ 0, где *a*22(1) – коэффициент, называемый *главным* (или *ведущим*) *элементом* 2-*го шага*. Вычислим множители 2-го шага

*qi*2 = *ai*2(1) / *a*22(1) (*i =*3, 4, …, *n*)

и вычтем последовательно из третьего, четвертого, …, *n-*го уравнения системы второе уравнение, умноженное соответственно на *q*32,*q*42, …,*qm*2.

В результате получим систему:

*a*11*x*1 + *a*12*x*2 + *a*13*x*3 + … + *a*1*nxn*= *b*1,

*a*22(1)*x*2 + *a*23(1)*x*3 + … + *a*2*n*(1) = *b*2(1),

*a*33(2)*x*3+ … + *a*3*n*(2)*xn* = *b*3(2),

. . . . . . . . . . . . . . . . . . .

*an*3(2)*x*3 + … + *ann*(2)*xn*= *bn*(2).

Здесь коэффициенты *aij*(2) и *bij*(2) вычисляются по формулам

*aij*(2) = *aij*(1) – *qi*2*a*2*j*(1) , *bi*(2) = *bi*(1) – *qi*2*b*2(1).

Аналогично проводятся остальные шаги. Опишем очередной *k-*й шаг.

*k-*й шаг. В предположении, что *главный*(*ведущий*) *элемент k-*го шага *akk*(*k*–1) отличен от нуля, вычислим *множители k-го шага*

*qik = aik*(*k*–1) / *ak*(*k*–1) (*i = k*+ 1, …, *n*)

и вычтем последовательно из (*k* + 1)-го, …, *n*-го уравнений полученной на предыдущем шаге системы *k*-e уравнение, умноженное соответственно на *qk*+1,*k*, *qk*+2,*k*, …, *qnk*.

После (*n -*1)-го шага исключения получим систему уравнений

*a*11*x*1 + *a*12*x*2 + *a*13*x*3 + … + *a*1*nxn* = *b*1,

*a*22(1)*x*2 + *a*23(1)*x*3 + … + *a*2*n*(1)*xn* = *b*2(1),

*a*33(2)*x*3 + … + *a*3*n*(2)*xn* = *b*3(2),

. . . . . . . . . . . . . . . . . . . .

*ann*(*n*–1)*xn* = *bn*(*n*–1).

матрица ***A***(*n*-1) которой является верхней треугольной. На этом вычисления прямого хода заканчиваются.

Обратный ход. Из последнего уравнения системы находим xn. Подставляя найденное значение *x* предпоследнее уравнение, получим *xn*–1. Осуществляя обратную подстановку, далее последовательно находим *xn*–1, *xn*–2, …, *x*1. Вычисления неизвестных здесь проводятся по формулам

*xn* = *bn*(*n*–1) / *ann*(*n*–1),

*xk* = (*bn*(*k*–1) – *ak*,*k*+1(*k*–1)*xk*+1 – … – *ak*(*k*–1)*xn*) / *akk*(*k*–1), (*k* = *n* – 1, …, 1).

Необходимость выбора главных элементов. Заметим, что вычисление множителей, а также обратная подстановка требуют деления на главные элементы *akk*(*k*–1). Поэтому если один из главных элементов оказывается равным нулю, то схема единственного деления не может быть реализована. Здравый смысл подсказывает, что и в ситуации, когда все главные элементы отличны от нуля, но среди них есть близкие к нулю, возможен неконтролируемый рост погрешности.

**2. Метод Гаусса с выбором главного элемента по столбцу (схема частичного выбора).**

  Описание метода. На *k*-м шаге прямого хода коэффициенты уравнений системы с номерами *i*= *k*+ 1, …, *n* преобразуются по формулам

*aij*(*k*) = *aij*(*k*–1)*− qikakj*, *bi*(*k*) = *bi*(*k*–1)*− qikbk*(*k*–1), *i* = *k* + 1, …, *n*.

Интуитивно ясно, что во избежание сильного роста коэффициентов системы и связанных с этим ошибок нельзя допускать появления больших множителей *qik*.

В методе Гаусса с выбором главного элементоа по столбцу гарантируется, что |*qik*| ≤ 1 для всех *k*= 1, 2, …, *n* – 1 и *i* = *k* + 1, …, *n*. Отличие этого варианта метода Гаусса от схемы единственного деления заключается в том, что на *k*-м шаге исключения в качестве главного элемента выбирают максимальный по модулю коэффициент *aik* при неизвестной *xk* в уравнениях с номерами *i = k* + 1, …, *n*. Затем соответствующее выбранному коэффициенту уравнение с номером *ik* меняют местами с *k*-м уравнением системы для того, чтобы главный элемент занял место коэффициента a(*k*-1). После этой перестановки исключение неизвестного *xk* производят, как в схеме единственного деления.

**3. Метод Гаусса с выбором главного элемента по всей матрице (схема полного выбора).**

В этой схеме допускается нарушение естественного порядка исключения неизвестных.

На 1-м шаге метода среди элементов *aij* определяют максимальный по модулю элемент *ai*1*j*1. Первое уравнение системы и уравнение с номером *i*1 меняют местами. Далее стандартным образом производят исключение неизвестного *xi*1 из всех уравнений, кроме первого.

На *k*-м шаге метода среди коэффициентов *aij*(*k*–1) при неизвестных в уравнениях системы с номерами *i* = *k*, …, *n* выбирают максимальный по модулю коэффициент *aij*(*k*-1). Затем *k*-е уравнение и уравнение, содержащее найденный коэффициент, меняют местами и исключают неизвестное *xjk* из уравнений с номерами *i* = *k* + 1, …, *n*.

На этапе обратного хода неизвестные вычисляют в следующем порядке: *xjn, xjn–*1*, …, xj*1.

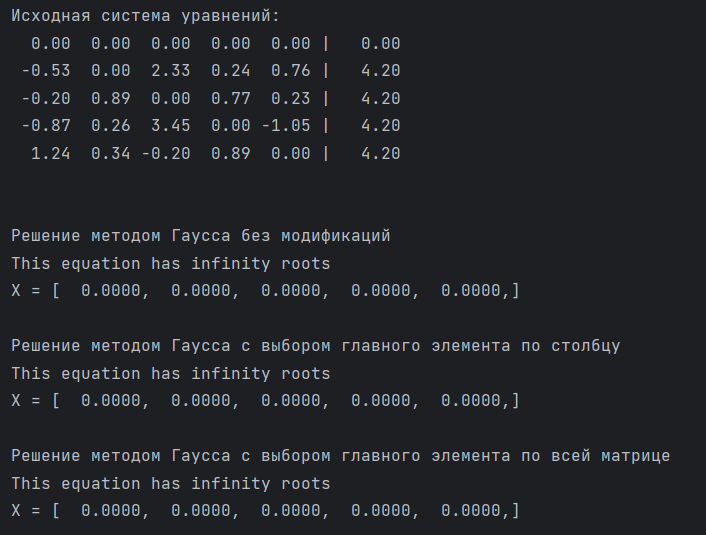
# Решение тестовых примеров

**Вариант 29.**

**Тестовый пример 1**

Методом Гаусса найти с точностью 0,0001 численное решение СЛАУ:

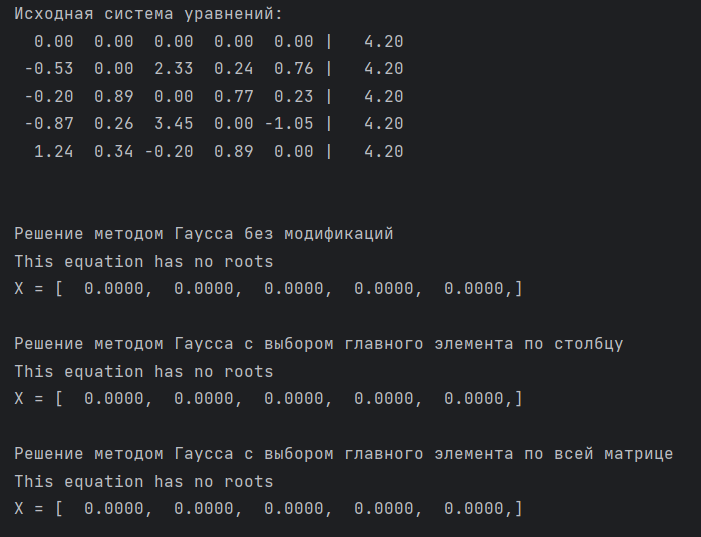
В следующей СЛАУ решением будет бесконечное количество решений, и как видно в выводе на консоль, программная реализация решает данную задачу.



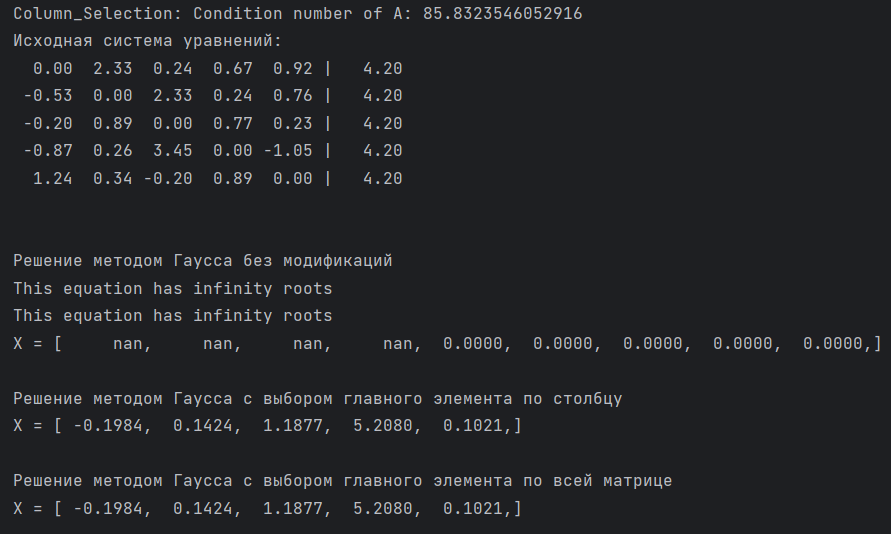
**Тестовый пример 2:**

Методом Гаусса найти с точностью 0,0001 численное решение СЛАУ:

В следующей СЛАУ решением будет отсутствие решений, и как видно в выводе на консоль, программная реализация решает данную задачу.



**Тестовый пример 3:**



Методом Гаусса найти с точностью 0,0001 численное решение СЛАУ:

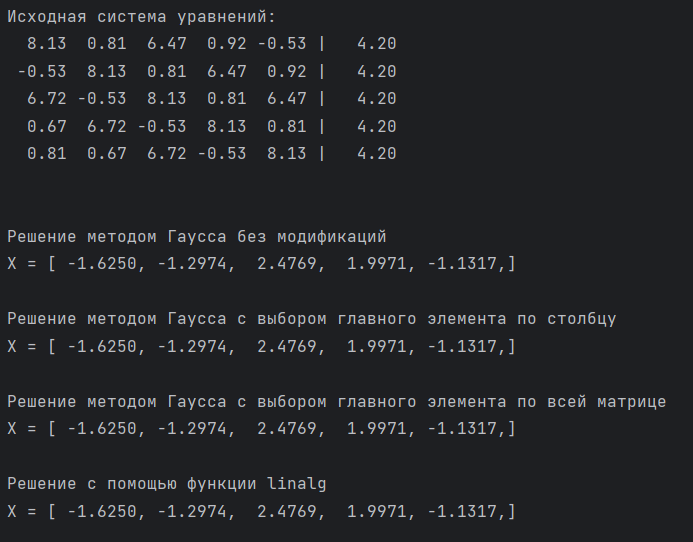
В следующей СЛАУ решением будет найдено единственное решение, несмотря на то что в матрице находятся нули на главной диагонали. Проверка решения происходит с помощью встроенной функции numpy.linalg и решение совпадает. Однако решение данной СЛАУ классическим методом гаусса невозможно осуществить, т.к. осуществляется деление на 0, что приводит к неопределенному поведению и появлению ошибок.

Выведем число обусловленности:



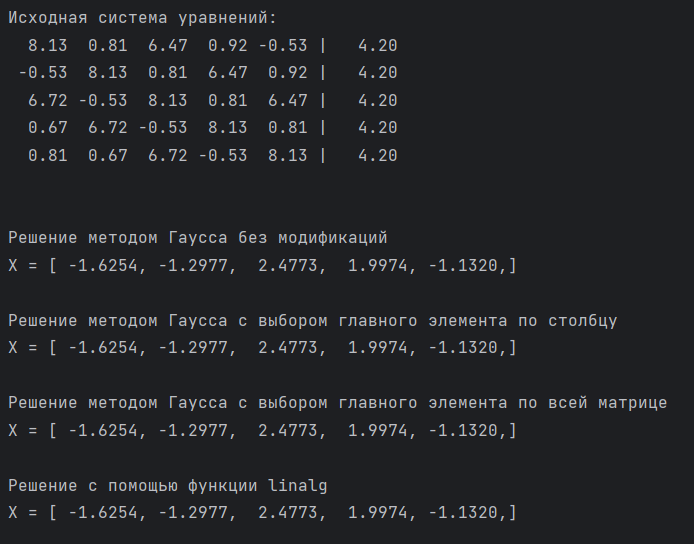


**Тестовый пример 4 из условия:**

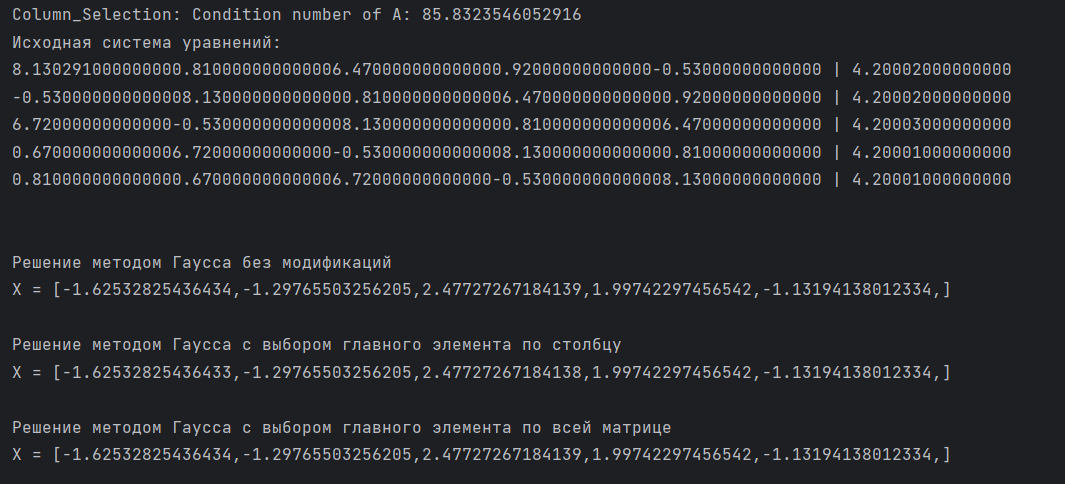


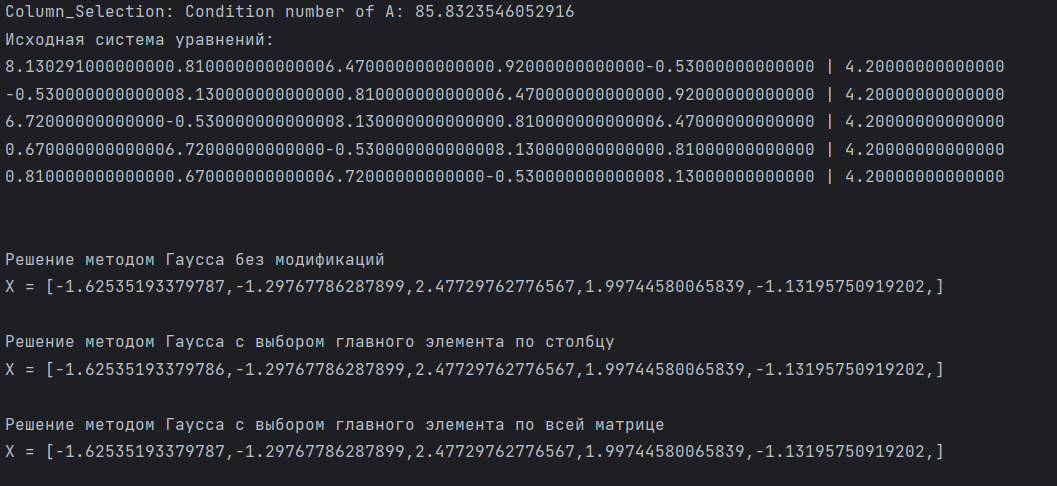
**Тестовый пример 5,** в котором одно значение из матрицы взято с значением на 0.000001 больше, чем в тестовом примере 4:

На скриншоте из консоли этого не видно, однако можно заметить, что решение СЛАУ изменились примерно на 0.0003, что говорит о высокой чувствительности матрицы и большом значении числа обусловленности.



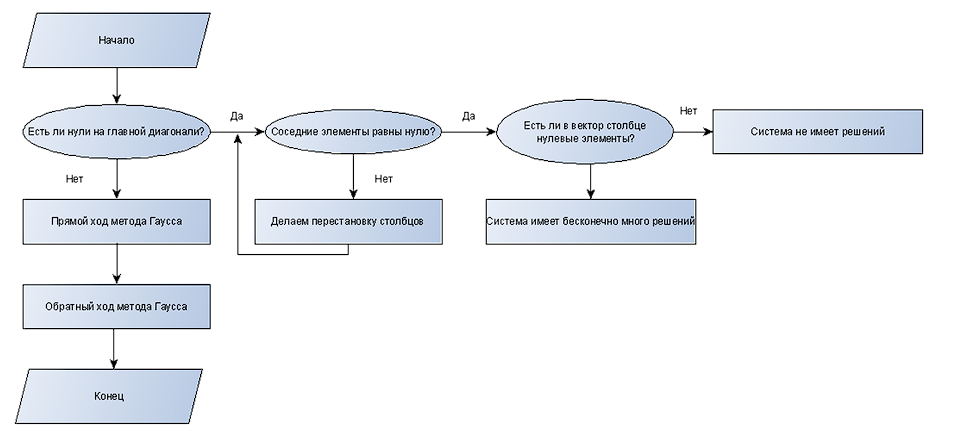
**Тестовый пример 6**  **из условия:**

****

****

Разность значений в данном примере ~ 0,00003.

1. Блок схема для алгоритма решения СЛАУ методом Гаусса



1. Выводы

В ходе выполнения данной лабораторной работы были решены некоторые примеры СЛАУ с помощью собственной программной реализации метода Гаусса и его модификаций. Данный метод является одним из самых распространенных и простых методов решения систем линейных уравнений (также существуют и более сложные методы, такие как метод Зейделя, метод простых итераций, метод квадратного корня и др).

Однако данный метод не является идеальным для решения СЛАУ при помощи вычислительных систем, ввиду следующих ошибок связанных с округлением чисел:

**Представление чисел с плавающей запятой:** В большинстве компьютерных систем числа представлены в форме чисел с плавающей запятой (например, в формате IEEE 754). Это представление ограничивает точность чисел, так как они имеют ограниченное количество бит для хранения дробных значений. При выполнении арифметических операций (сложение, вычитание, умножение, деление) над числами с плавающей запятой могут возникать ошибки округления.

**Потеря значащих битов**: При выполнении множественных арифметических операций над числами с плавающей запятой, особенно при вычитании чисел, могут потеряться значащие биты, что приводит к потере точности. Это называется ошибкой "катастрофической потери".

**Масштабирование и нормализация:** В методе Гаусса для приведения матрицы к верхне-треугольному виду (или к другим формам), производятся множественные операции умножения и деления строк на константы. Эти операции могут привести к большим или очень маленьким значениям в матрице, что может сказаться на точности.

**Матрицы с плохой обусловленностью:** Если матрица системы близка к вырожденной (матрице с плохой обусловленностью), то метод Гаусса и другие методы численного анализа могут усугубить ошибки округления. Матрицы с плохой обусловленностью могут привести к неустойчивости численных методов.

При решении тестовых примеров стало ясно, что метод Гаусса справляется с СЛАУ с бесконечным количеством решений (выявляет их), c отсутствием решений, с матрицами с нулевой главной диагональю (классический метод Гаусса не справляется, но методы с модификациями с легкостью).

Также было проведено исследование в ходе которого было выяснено, что при изменении одного элемента матрицы на 1\*10^(-6), результат изменился с точностью на 0,0003, что говорит о высокой чувствительности матрицы и большом значении числа обусловленности.