Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра информатики

Отчет по лабораторной работе №1

Решение систем линейных алгебраических уравнений (СЛАУ)   
методом Гаусса и с помощью его модификаций

Выполнил:

студент группы 053506

Ермолович Д.C

Руководитель:

доцент

Анисимов В.Я.

Минск

**Содержание**

[**Цель работы** 3](#_Toc96158677)

[**Краткие теоретические сведения** 3](#_Toc96158678)

[Метод Гаусса 3](#_Toc96158679)

[**1. Схема единственного деления.** 4](#_Toc96158680)

[**2. Метод Гаусса с выбором главного элемента по столбцу (схема частичного выбора).** 5](#_Toc96158681)

[**3. Метод Гаусса с выбором главного элемента по всей матрице (схема полного выбора).** 6](#_Toc96158682)

[**Несовместные системы** 6](#_Toc96158683)

[**Программная реализация** 8](#_Toc96158684)

[**Код** 8](#_Toc96158685)

[**Тесты** 11](#_Toc96158686)

[**Вывод** 13](#_Toc96158687)

# **Цель работы**

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

# **Краткие теоретические сведения**

СЛАУ обычно записывается в виде

, или коротко Ax=b, где

A = ; a = ; b = .

Здесь А и b заданы и требуется найти x.

Методы решения СЛАУ делятся на прямые и итерационные.

Прямые методы дают в принципе точное решение за конечное число арифметических операций. Они просты и наиболее универсальны. Для хорошо обусловленной системы необходимого порядка n 200 применяются практически только прямые методы.

Наибольшее распространение среди методов получил метод Гаусса и его модификации.

## 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),

Начало формы

Конец формы

*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) / *akk*(*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 *xn* в 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 – … – *akn*(*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 оказывыется 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 *aikk* 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*akk*(*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 мтода 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 *aikjk*(*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.

# **Несовместные системы**

При решении СЛАУ не всегда встречаются совместные системы. На практике широко распространены еще два случая:

– Система несовместна (не имеет решений);  
– Система совместна и имеет бесконечно много решений.

1. После применения метода Гаусса последнее уравнение, в котором хотя бы один коэффициент отличен от нуля окажется равным одному из трех типов:
   * 1. A: 0\*+0\*+0\*+0\*=7, нет решений(Система несовместна)
     2. B: 0\*+0\*+0\*+5\*=7, хотя бы одно решение(Система совместна)
     3. C: 0\*+3\*+2\*+5\*=7, бесконечно много решений(Система совместна и имеет бесконечно много решений)
2. Если главный определитель равен нулю и хотя бы один их вспомогательных определителей отличен от нуля, то система решений не имеет. (Система несовместна).
3. Если главный определитель и все вспомогательные определители равны нулю, то система имеет бесконечно много решений.( Система совместна и имеет бесконечно много решений).
4. Если определитель не равен нулю, система имеет единственное [решение](https://lms2.sseu.ru/courses/eresmat/gloss/g98.htm)(Система совместна).
5. Если количество уравнений меньше чем количество переменных, то система несовместна или имеет бесконечно много решений.
6. Если число уравнений больше числа неизвестных, то система переопределенная. Решить данную систему можно методом наименьших квадратов.
7. На главной диагонали не могут быть нули, так как при вычислении коэффициента в знаменателе будет 0.

# 

# **Программная реализация**

## **Код**

def check\_compatibility(A, b):  
 n, m = A.shape  
 if n < m:  
 print("The system is consistent and has infinitely many solutions")  
 return 0  
 elif n > m:  
 print("The system is overridden i can't solve it")  
 return 0  
  
 main\_det = np.linalg.det(A)  
 dets = []  
 for i in range(2):  
 A\_copy = copy.deepcopy(A)  
 A\_copy[:, i] = b\_2  
 dets.append(np.linalg.det(A\_copy))  
 if main\_det == 0:  
 for elem in dets:  
 if elem != 0:  
 print("Has no solutions")  
 return 0  
 print("Infinitely many solutions")  
 return 0  
 return 1

Проверка матрицы на несовместность. Здесь проверяются все случаи, которые были описаны в теме несовместные системы.

def check\_diagonal(A):  
 n = A.shape[0]  
 stack = []  
 for i in range(n):  
 if A[i][i] == 0:  
 stack.append(i)  
  
 if len(stack) == 0:  
 return 1  
  
 for i in stack:  
 for j in range(n):  
 if A[j, i] != 0 and A[i, j] != 0:  
 buf = copy.deepcopy(A[j, :])  
 A[j, :] = A[i, :]  
 A[i, :] = buf  
 break  
 stack.clear()  
  
 for i in range(n):  
 for j in range(n):  
 if A[i][i] == 0:  
 stack.append(i)  
  
 if len(stack) != 0:  
 return 0  
 else:  
 return 1

Устранение нулей на диагонали.

def print\_answer(A, x, b):  
 n = A.shape[0]  
 for i in range(n):  
 print(f"x{i} = {x[i]}")  
 print(f"\nOriginal b = {b}")  
 print(f"\nCheck array: b = {np.dot(A, x)}")

Проверка решения

def first\_way(A):  
 # Приводим А к нижнетреугольной, прямой ход  
 n = A.shape[0]  
 for i in range(n - 1):  
 for j in range(i + 1, n, 1):  
 q = A[j, i] / A[i, i]  
 A[j, :] = A[j, :] - q \* A[i, :]  
 # Обратный ход  
 x = np.array(np.zeros(n))  
 kof = 0  
 for i in range(n - 1, -1, -1):  
 kof = np.dot(x, A[i, :n])  
 x[i] = (A[i, n] - kof) / A[i, i]  
  
 return x, n

Метод единственного деления

def second\_way(A):  
 n = A.shape[0]  
  
 # Прямой ход  
 for i in range(n - 1):  
  
 elem = np.max(A[i:n, i])  
 index = np.where(A[i:n, i] == elem)  
  
 buf = copy.deepcopy(A[index[0] + i, :])  
 A[index[0] + i, :] = A[i, :]  
 A[i, :] = buf  
  
 for j in range(i + 1, n, 1):  
 q = A[j, i] / A[i, i]  
 A[j, :] = A[j, :] - q \* A[i, :]  
  
 # Обратный ход  
 x = np.array(np.zeros(n))  
 kof = 0  
 for i in range(n - 1, -1, -1):  
 kof = np.dot(x, A[i, :n])  
 x[i] = (A[i, n] - kof) / A[i, i]  
 return x, n

Метод частного выбора

def third\_way(A):  
 n = A.shape[0]  
 stack = []  
 for i in range(n - 1):  
  
 coord = unravel\_index(A[i:n, i:n].argmax(), A[i:n, i:n].shape)  
 index, row = coord  
 stack.append(row + i)  
  
 buf = copy.deepcopy(A[index + i, :])  
 A[index + i, :] = A[i, :]  
 A[i, :] = buf  
  
 for j in range(i + 1, n, 1):  
 q = A[j, row + i] / A[i, row + i]  
 A[j, :] = A[j, :] - q \* A[i, :]  
  
 sort\_stack = np.sort(stack)  
 ans = 0  
 for i in range(n - 1):  
 if sort\_stack[i] != i:  
 ans = i  
 break  
 stack.append(ans)  
  
 x = np.array(np.zeros(n))  
 for i in range(n - 1, -1, -1):  
 index = stack.pop()  
 s = 0  
 for j in range(n):  
 if j != index:  
 s = s + x[j] \* A[i, j]  
 x[index] = (A[i, n] - s) / A[i, index]  
 return x, n

Метод полного выбора

def error\_estimate(x, A, b, b\_2):  
 b\_t = np.dot(A, x)  
 x\_copy = copy.deepcopy(x)  
 delta\_b = b\_t - b\_2  
  
 A\_expand = np.zeros((A.shape[0], A.shape[0] + 1))  
 A\_expand[:, :A.shape[0]] = A  
 A\_expand[:, A.shape[0]] = delta\_b  
  
 delta\_x, n = first\_way(A\_expand)  
  
 for i in range(len(delta\_x)):  
 delta\_x[i] = abs(delta\_x[i])  
 absolute = max(delta\_x)  
  
 for i in range(len(x\_copy)):  
 x\_copy[i] = abs(x\_copy[i])  
  
 relative = absolute / max(x\_copy)  
 return absolute, relative

Оценка погрешности

## **Тесты**

1.

A = np.array([  
 [3, -4],  
 [3, -4]  
])  
b = np.array([[12], [18]])

A\*x=b

Вывод: Не имеет решений

2.

A = np.array([  
 [7, 6],  
 [ 3.5,3]  
 ])  
b = np.array([[-42], [-21]])

A\*x=b

Вывод: Бесконечно много решений

3.

A = np.array([  
 [7, 6, 6],  
 [3.5, 3, 3]  
])  
b = np.array([[-42], [-21]])

A\*x=b

Вывод: Система неопределенна или имеет бесконечно много решений

4.

A = np.array([  
 [7, 6],  
 [3.5, 3],  
 [3.5, 3]  
])  
b = np.array([[-42], [-21], [43]])

A\*x=b

Вывод: Система переопределена.

5.

A = np.array([  
 [2.33, 0.81, 0.67],  
 [-0.53, 0, 1],  
 [0.92, -0.53, 0]  
])  
b = np.array([[4], [21], [9]])

A\*x=b

Ответ: x= [0.9, -15.42, 21.48] (ответ получен с точностью <10^-4)

**Вариант 10**

Методом Гаусса найти с точностью 0,0001 численное решение системы **Ax=b**:

Ответ:

|  |  |  |
| --- | --- | --- |
| Схема единственного деления | Схема частичного выбора | Схема полного выбора |
|  |  |  |

# **Вывод**

В ходе этой лабораторной работе я изучил метод Гаусса и его модификации.

Составил программу, которая решает СЛАУ любым из выбранных способов.

Отработал несовместные матрицы системы.

Выполнили тестовые примеры и проверили правильность работы программы.