Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА №1**

по дисциплине

‘ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА’

Вариант №19

*Выполнил:*

Студент группы P3213

Свиридов Дмитрий Витальевич

*Преподаватель:*

Малышева Татьяна Алексеевна



Санкт-Петербург, 2021

**Цель работы**

Изучить численные методы решения систем линейных алгебраических уравнений и реализовать один из них средствами программирования.

**Описание метода**

Метод Гаусса с выбором главного элемента по столбцам.

Схема с выбором главного элемента является одной из модификаций метода Гаусса. Идеей является такая перестановка уравнений, чтобы на k-ом шаге исключения ведущим элементом 𝑎𝑖𝑖 оказывался наибольший по модулю элемент k-го столбца.

**Листинг программы**

def solve\_minor(matrix, i, j):

""" Найти минор элемента матрицы """

n = len(matrix)

return [[matrix[row][col] for col in range(n) if col != j] for row in range(n) if row != i]

def solve\_det(matrix):

""" Найти определитель матрицы """

n = len(matrix)

if n == 1:

return matrix[0][0]

det = 0

sgn = 1

for j in range(n):

det += sgn \* matrix[0][j] \* solve\_det(solve\_minor(matrix, 0, j))

sgn \*= -1

return det

def solve(matrix):

""" Метод Гаусса с выбором главного элемента по столбцам """

n = len(matrix)

det = solve\_det([matrix[i][:n] for i in range(n)])

if det == 0:

return None

# Прямой ход

for i in range(n - 1):

# Поиск максимального элемента в столбце

max\_i = i

for m in range(i + 1, n):

if abs(matrix[m][i]) > abs(matrix[max\_i][i]):

max\_i = m

# Перестановка строк

if max\_i != i:

for j in range(n + 1):

matrix[i][j], matrix[max\_i][j] = matrix[max\_i][j], matrix[i][j]

# Исключение i-того неизвестного

for k in range(i + 1, n):

coef = matrix[k][i] / matrix[i][i]

for j in range(i, n + 1):

matrix[k][j] -= coef \* matrix[i][j]

reduced\_matrix = matrix[:]

# Обратный ход

roots = [0] \* n

for i in range(n - 1, -1, -1):

s\_part = 0

for j in range(i + 1, n):

s\_part += matrix[i][j] \* roots[j]

roots[i] = (matrix[i][n] - s\_part) / matrix[i][i]

# Вычисление невязок

residuals = [0] \* n

for i in range(n):

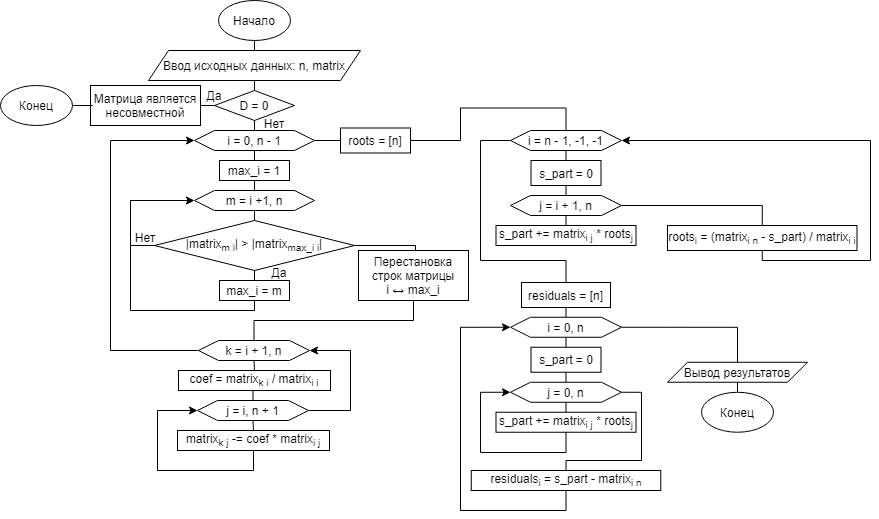
s\_part = 0

for j in range(n):

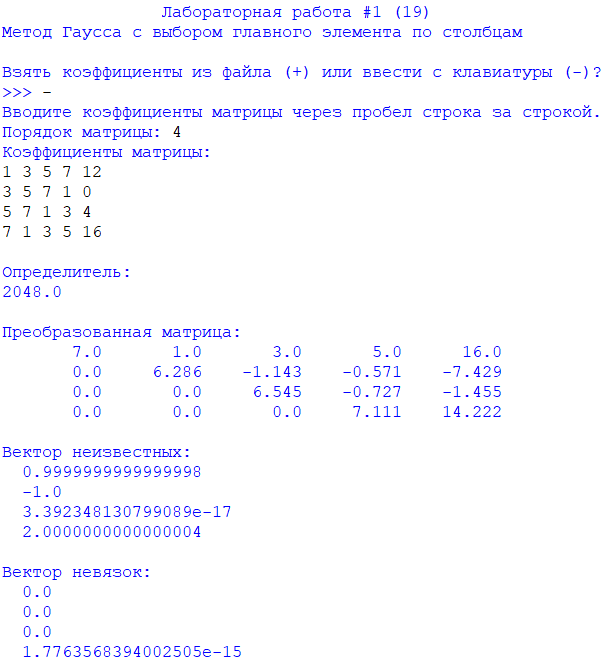
s\_part += matrix[i][j] \* roots[j]

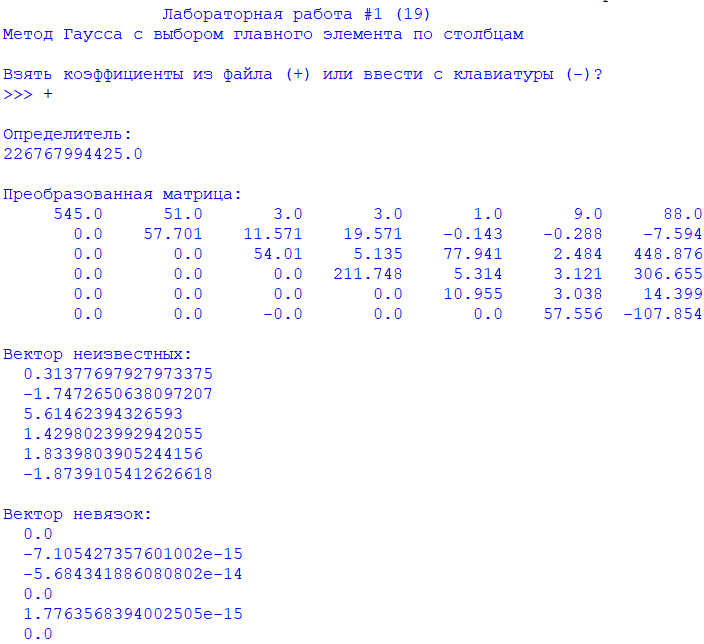
residuals[i] = s\_part - matrix[i][n]

return det, reduced\_matrix, roots, residuals

**Блок-схема метода**

**Примеры работы программы**





**Вывод**

В результате выполнения данной лабораторной работой я познакомился с численными методами решения математических задач на примере систем алгебраических уравнений, реализовав на языке программирования Python метод Гаусса с выбором главного элемента по столбцам.