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: **2**

Преподаватель:   
Малышева Татьяна Алексеевна

Выполнил: Амузинский Артем

Группа: Р3206

Санкт-Петербург, 2025г

# Цель работы

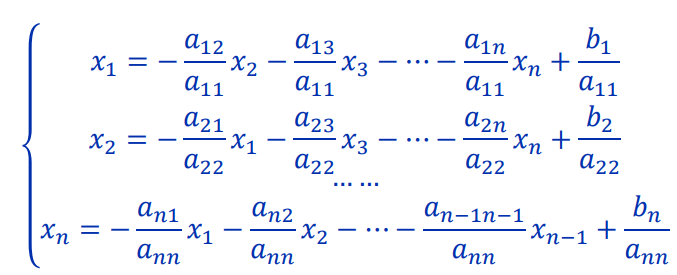
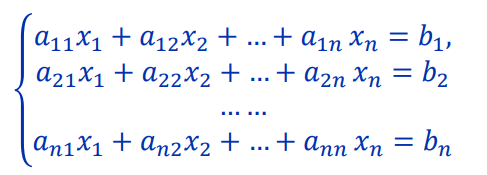
Изучить численные методы решения систем линейных алгебраических уравнений и реализовать один из них средствами программирования.

# Описание метода

Итерационны метод сводится к тому, что нам надо найти такой предел, при котором вектор иксов будет стремится к точному решению слау.

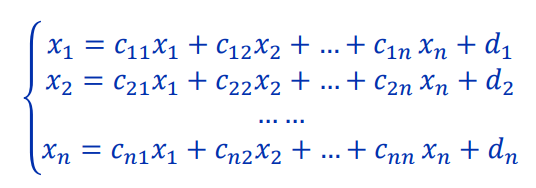
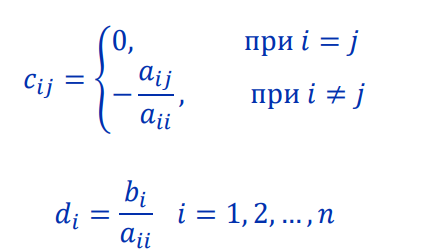
Данная программа находит решение слау при помощи метода простых итераций, который заключается в приблиенном нахождении решения.

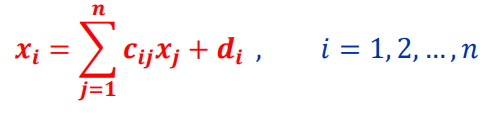
1. выражаем x в каждой строке матрицы



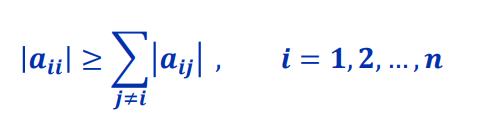
2) заменяем коэффициенты перед х на c\_(i, j) где при i = j: с\_(i, j) = 0 при такой замаен

матрицу можно представить x\_i = сумма от первой строчки до последней от суммы произведений всех коэффициентов и иксов плюс свободный коэффициент

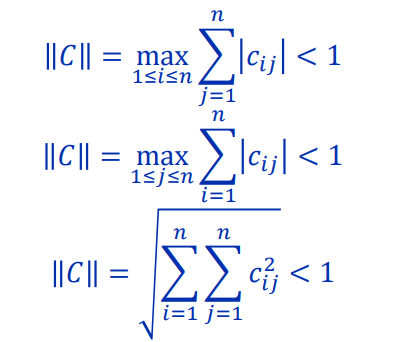




3) проверка на теорему о преобладании диагональных элементов



4) проверка на теорему о норме матрицы С



5) далее эти условия повторяются в цикле, лиюо пока не будет итераций более, чем задано в начале, либо когда не будет достигнута нужная точность

# Листинг программы

def sordDiag():  
 for i in range(h):  
 elements = []  
 for j in range(h):  
 if j < i:  
 elements.append(0)  
 else:  
 elements.append(abs(matrix[j][i]))  
 matrix[i], matrix[elements.index(max(elements))] = matrix[elements.index(max(elements))], matrix[i]  
  
  
def firstTheorem():  
 for i in range(h):  
 if 2 \* abs(matrixC[i][i]) < abs(sum(matrixC[i])):  
 return False  
 return True  
  
  
def toDiag():  
 for i in range(h):  
 delitel = matrixC[i][i]  
 for j in range(h):  
 if i == j:  
 matrixC[i][j] = 0  
 else:  
 matrixC[i][j] /= -delitel  
 matrixD[i] /= delitel  
  
  
def secondTheorem():  
 for i in range(h):  
 if abs(sum(matrixC[i])) <= eps:  
 print("False")  
 return False  
 return True  
  
  
def printMatrix():  
 for i in range(h):  
 print(matrix[i])  
  
  
def makeMatrixs():  
 for i in range(h):  
 forC = []  
 for j in range(h):  
 forC.append(matrix[i][j])  
 matrixC.append(forC)  
 matrixD.append(matrix[i][-1])  
  
  
def start():  
 for i in range(h):  
 matrix.append([float(j) for j in input().split()])  
  
 print("Исходная матрица:")  
 printMatrix()  
  
 sordDiag()  
  
 print("Преобразованная матрица:")  
 printMatrix()  
  
 makeMatrixs()  
  
 if not firstTheorem():  
 print("Диагональное преобладание невозможно!")  
 return False  
  
 toDiag()  
  
 if not secondTheorem():  
 print("Метод не сходится!")  
 return False  
  
 matrixK = []  
 prevMatrixK = [i for i in matrixD]  
 iterations = 1  
  
 while iterations < max\_iter:  
 for i in range(h):  
 row = 0  
 for j in range(h):  
 row += matrixC[i][j] \* prevMatrixK[j]  
 row += matrixD[i]  
 matrixK.append(row)  
  
 delt = 0  
 for i in range(h):  
 delt = max(abs(matrixK[i] - prevMatrixK[i]), delt)  
  
 if delt < eps:  
 print(f"Решение найдено за {iterations} итераций")  
 print("Вектор неизвестных:", matrixK)  
 print("Вектор погрешностей:", [abs(matrixK[i] - prevMatrixK[i]) for i in range(h)])  
 return True  
 else:  
 prevMatrixK = matrixK  
 matrixK = []  
 iterations += 1  
  
 print("Достигнуто максимальное число итераций. Решение могло не сойтись.")  
 return False  
  
  
eps = float(input("Введите точность: "))  
max\_iter = int(input("Введите максимальное число итераций: "))  
h = int(input("Введите размерность матрицы: "))  
  
matrix = []  
matrixC = []  
matrixD = []  
  
if start():  
 print("Метод успешно завершился.")  
else:  
 print("Решение не найдено.")

так же программу можно посмотреть на сайте гитхаб:

<https://github.com/listoCheck/Vichmat/tree/main/lab1>

# Примеры работы программы

|  |  |  |
| --- | --- | --- |
| № | ввод | вывод |
| 1 | Введите точность: 0.01  Введите максимальное число итераций: 15  Введите размерность матрицы: 3  2 2 10 14  10 1 1 12  2 10 1 13 | **Пример из лекции, ответ (1, 1, 1) сошлось**  Исходная матрица:  [2.0, 2.0, 10.0, 14.0]  [10.0, 1.0, 1.0, 12.0]  [2.0, 10.0, 1.0, 13.0]  Преобразованная матрица:  [10.0, 1.0, 1.0, 12.0]  [2.0, 10.0, 1.0, 13.0]  [2.0, 2.0, 10.0, 14.0]  Решение найдено за 5 итераций  Вектор неизвестных: [0.999568, 0.99946, 0.9993159999999999]  Вектор погрешностей: [0.0019320000000000448, 0.0024600000000001288, 0.0030840000000000867]  Метод успешно завершился. |
| 2 | Введите точность: 0.01  Введите максимальное число итераций: 10  Введите размерность матрицы: 2  2 1 4  1 6 13 | **Пример придумал сам на бумаге, ответ (1, 2) сошлось**  Исходная матрица:  [2.0, 1.0, 4.0]  [1.0, 6.0, 13.0]  Преобразованная матрица:  [2.0, 1.0, 4.0]  [1.0, 6.0, 13.0]  Решение найдено за 5 итераций  Вектор неизвестных: [0.9994212962962963, 1.9988425925925923]  Вектор погрешностей: [0.007523148148148362, 0.0023148148148151027]  Метод успешно завершился. |
| 3 | Введите точность: 0.001  Введите максимальное число итераций: 15  Введите размерность матрицы: 3  1.1 -0.2 0.1 1.6  0.1 -1.2 -0.2 2.3  0.2 -0.1 1.1 1.5 | **Пример взял из интернета, ответ (1, -2, 1) сошлось**  [clck.ru/3GZBoc](https://clck.ru/3GZBoc" \t "_blank)  Исходная матрица:  [1.1, -0.2, 0.1, 1.6]  [0.1, -1.2, -0.2, 2.3]  [0.2, -0.1, 1.1, 1.5]  Преобразованная матрица:  [1.1, -0.2, 0.1, 1.6]  [0.1, -1.2, -0.2, 2.3]  [0.2, -0.1, 1.1, 1.5]  Решение найдено за 5 итераций  Вектор неизвестных: [1.0000443072835514, -1.999946445535886, 1.0000142020072817]  Вектор погрешностей: [8.613323378514615e-05, 1.901571551865544e-05, 0.00035644176708193864]  Метод успешно завершился. |
| 4 | Введите точность: 0.01  Введите максимальное число итераций: 25  Введите размерность матрицы: 2  2 2  2 2 | **Пример придумал сам, чтобы код не смог найти решение, сошлось**  Исходная матрица:  [2.0, 2.0]  [2.0, 2.0]  Преобразованная матрица:  [2.0, 2.0]  [2.0, 2.0]  Достигнуто максимальное число итераций. Решение могло не сойтись.  Решение не найдено. |

# 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 простых итераций.