Университет ИТМО

Лабораторная работа №1

по вычислительной математике

Вариант #18

Выполнил

Студент группы Р3212

Шаматульский Роман Константинович

Преподаватель

Машина Екатерина Алексеевна

Санкт-Петербург, 2025

Оглавление

[Цель работы 3](#_Toc190978930)

[Описание метода 4](#_Toc190978931)

[Листинг решения с помощью python 5](#_Toc190978932)

[Результат работы программы 6](#_Toc190978933)

[Вывод 7](#_Toc190978934)

# Цель работы

Реализовать с помощью программирования метод решения СЛАУ Гаусса-Зейделя.

# Описание метода

Метод Гаусса-Зейделя — итерационный численный метод решения СЛАУ, который улучшает метод простых итераций. Он использует разбиение матрицы на диагональную, нижнетреугольную и верхнетреугольную части и на каждом шаге обновляет значения неизвестных сразу после их вычисления.

Формула нахождения каждой *i-ой* компоненты *(k+1)* приближения:

# Листинг программы python

import numpy as np  
  
def gauss\_seidel(A, b, tol, max\_iter=10000):  
 n = len(b)  
 x = np.zeros(n)  
 for k in range(1, max\_iter+1):  
 x\_old = x.copy()  
 for i in range(n):  
 s = np.dot(A[i, :i], x[:i]) + np.dot(A[i, i+1:], x\_old[i+1:])  
 x[i] = (b[i] - s)/A[i, i]  
 err = np.abs(x - x\_old)  
 if np.all(err < tol):  
 return x, k, err  
 return x, k, err  
  
def make\_diagonally\_dominant(A, b):  
 n = A.shape[0]  
 for i in range(n):  
 for j in range(i, n):  
 if abs(A[j, i]) >= np.sum(np.abs(A[j, :])) - abs(A[j, i]):  
 if j != i:  
 A[[i, j]], b[[i, j]] = A[[j, i]].copy(), b[[j, i]].copy()  
 break  
 for i in range(n):  
 if abs(A[i, i]) < np.sum(np.abs(A[i, :])) - abs(A[i, i]):  
 return False  
 return True  
  
def read\_matrix():  
 mode = input("Чтение матрицы из файла? (y/n): ")  
 if mode.lower() == "y":  
 fn = input("Имя файла: ")  
 data = np.loadtxt(fn)  
 A = data[:, :-1]  
 b = data[:, -1]  
 else:  
 n = int(input("n: "))  
 A = np.zeros((n, n))  
 b = np.zeros(n)  
 for i in range(n):  
 row = list(map(float, input(f"Строка {i+1} (n+1 чисел): ").split()))  
 A[i, :] = row[:-1]  
 b[i] = row[-1]  
 return A, b  
  
def read\_tol():  
 mode = input("Чтение точности из файла? (y/n): ")  
 if mode.lower() == "y":  
 fn = input("Имя файла с точностью: ")  
 tol = float(open(fn).read().strip())  
 else:  
 tol = float(input("Точность: "))  
 return tol  
  
def main():  
 A, b = read\_matrix()  
 tol = read\_tol()  
 if not make\_diagonally\_dominant(A, b):  
 print("Невозможно добиться диагонального преобладания")  
 return  
 x, iters, err = gauss\_seidel(A, b, tol)  
 norm = np.linalg.norm(A, ord=np.inf)  
 print("Норма матрицы (сумма по строкам, бесконечная):", norm)  
 print("Решение x:", x)  
 print("Число итераций:", iters)  
 print("Вектор погрешностей:", err)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

# Результат работы программы

Чтение матрицы из файла? (y/n): n

n: 3

Строка 1 (n+1 чисел): 4 1 1 7

Строка 2 (n+1 чисел): 1 3 1 -2

Строка 3 (n+1 чисел): 1 1 5 9

Чтение точности из файла? (y/n): n

Точность: 0.001

Норма матрицы (сумма по строкам, бесконечная): 7.0

Решение x: [ 1.76003346 -1.86002259 1.81999783]

Число итераций: 6

Вектор погрешностей: [3.68740355e-05 1.45561021e-04 3.64870113e-05]

# Вывод

В ходе выполнения данной лабораторной работы, я познакомился с различными методами решения СЛАУ и реализовал один из них на практике. Полученные навыки помогут мне в решении математических задач в будущем.