



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Дальневосточный федеральный университет»
(ДВФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Департамент математического и компьютерного
моделирования

ОТЧЕТ

к лабораторной работе №8 по дисциплине
«Вычислительная математика»

Направление подготовки
01.03.02 «Прикладная математика и информатика»

Выполнил студент
гр. Б9119-01.03.02 систпро
Нагорнов С.С.

(ФИО)

(Подпись)

« 20 » _____ июня 2021 г.

г. Владивосток
2021

Содержание

Введение	2
Постановка задачи	2
Алгоритм работы	2
Приложения	3
Вывод	4

Введение

В данной лабораторной работе необходимо найти решение СЛАУ с помощью методов Зейделя и Якоби.

Постановка задачи

Дана матрица A и вектор B . Необходимо решить систему $Ax = B$ с помощью итерационных методов Зейделя и Якоби.

Алгоритм работы

1. Для нахождения элементов решения x методом Якоби будем пользоваться следующей формулой:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1, j \neq i}^n a_{ij} x_j^{(k)} \right)$$

2. $x^{(0)}$ возьмем вектор размером n , которая состоит из нулей. Если $i = j$, сумма принимает значение 0; $i = \overline{1, n}$.
3. Для нахождения решения СЛАУ методом Зейделя используется другая формула:

$$x_i^{(k+1)} = b_i + \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} + \sum_{j=i+1}^n a_{ij} x_j^{(k)}, \quad i = \overline{1, n}$$

4. Используя данные методы, установим максимальное количество итераций. В нашем случае $k = 10$.

Приложения

```
1  import numpy as np
2
3
4  def jacobi_method(A, f, it=10):
5      n = len(f)
6      x_prev = x = np.array([0.0] * n)
7
8      for k in range(it):
9          for i in range(n):
10             S = sum([A[i, j] * x_prev[j] if i != j else 0 for j in
range(n)])
11             x[i] = (f[i] - S) / A[i, i]
12
13             x_prev = x
14
15     return x
16
17
18 def seidel_method(A, f, it=10):
19     n = len(f)
20     xp = x = np.array([0.0] * n)
21
22     for k in range(it):
23         for i in range(n):
24             S1 = sum([A[i, j] * x[j] for j in range(i)])
25             S2 = sum([A[i, j] * xp[j] for j in range(i + 1, n)])
26             x[i] = (f[i] - S1 - S2) / A[i, i]
27
28         xp = x
29
30     return x
31
32
33 def main():
34     mat = np.matrix([
35         [9.2, 2.5, -3.7],
36         [0.9, 9., 0.2],
37         [4.5, -1.6, -10.3]
38     ])
39     vec = np.array([-17.5, 4.4, -22.1])
40     print(jacobi_method(mat, vec))
41     print(seidel_method(mat, vec))
42
43
44 if __name__ == '__main__':
45     main()
```

Листинг 1: Компьютерная реализация алгоритма

Вывод

В данной лабораторной работе были реализованы итерационные методы Зейделя и Якоби для решения СЛАУ.