



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 №9 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 | 3 |

Введение

В данной лабораторной работе необходимо найти решение СЛАУ с помощью метода релаксации.

Постановка задачи

Дана матрица A и вектор B . Необходимо решить систему $Ax = B$ с помощью итерационного метода релаксации.

Алгоритм работы

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

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

2. $\overline{x^{(0)}}$ возьмем вектор размером n , которая состоит из нулей; $i = \overline{1, n}$.
3. Параметр ω определяет релаксацию. При $\omega > 1$ – метод верхней релаксации, при равном 1 – метод Зейделя, меньше 1 – метод нижней релаксации.
4. Используя метод, установим максимальное количество итераций. В нашем случае $k = 10$.

Приложения

```
1  import numpy as np
2
3
4  def relax_method(A, f, it=10):
5      n = len(f)
6      xp = x = np.array([0.0] * n)
7      w = 1.2
8
9      for k in range(it):
10         for i in range(n):
11             S1 = sum((A[i, j] * x[j] for j in range(i)))
12             S2 = sum((A[i, j] * xp[j] for j in range(i + 1, n)))
13             x[i] = (w * (f[i] - S1 - S2) + (1 - w) * A[i, i] * xp[i])
14                 / A[i, i]
15         xp = x
16
17     return x
18
19
20 def main():
21     mat = np.matrix([
22         [9.2, 2.5, -3.7],
23         [0.9, 9., 0.2],
24         [4.5, -1.6, -10.3]
25     ])
26     vec = np.array([-17.5, 4.4, -22.1])
27
28     print(relax_method(mat, vec))
29
30
31 if __name__ == '__main__':
32     main()
33
```

Листинг 1: Компьютерная реализация алгоритма

Вывод

В данной лабораторной работе был реализован итерационный метод релаксации для решения СЛАУ.