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 №10 по дисциплине
«Вычислительная математика»

Направление подготовки
01.03.02 «Прикладная математика и информатика»

Выполнил студент
гр. Б9119-01.03.02систпро
Нагорнов С.С.
(ФИО) *(Подпись)*
«20» июня 2021 г.

г. Владивосток
2021

Содержание

Введение

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

Постановка задачи

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

Алгоритм работы

1. Метод Ричардсона является ускоренным методом простых итераций за счет параметра t, вычисляемый по формуле:

$$t = \frac{2}{(m+M) + (M+m)\cos\left(\frac{(2i-1)\pi}{2k}\right)}$$

.

2. В вычислении параметра t присутствуют значения m и M. Данные значения являются собственными у матрицы A минимальным и максимальным соответственно.

3. Формула для вычисления решения имеет следующий вид:

$$x^{(k+1)} = t(B - Ax^{(k)})$$

.

4. Используя данный метод, установим максимальное количество итераций. В нашем случае $k = 16$.

Приложения

```
1  from numpy import linalg
2  import math
3
4
5  def richardson(mat, vec, k):
6    a, b = __find_min_max_eig(mat)
7    dim = len(vec)
8    x = [[0 for _ in range(dim)]]
9
10   it = 0
11   while it < k:
12     it += 1
13     x.append([])
14     t = 2 / ((a + b) + (b - a) * math.cos((2 * it - 1) * math.pi
   / (2 * k)))
15     for i in range(dim):
16       x[it].append(x[it - 1][i] -
17                __find_subtraction(x[it - 1], mat, vec, i) * t)
18
19   return x[-1], it
20
21
22 def __find_min_max_eig(mat):
23   eig = linalg.eig(mat)[0]
24   return min(eig), max(eig)
25
26
27 def __find_subtraction(x, mat, vec, pos):
28   sub = -vec[pos]
29   for j in range(len(vec)):
30     sub += mat[pos][j] * x[j]
31   return sub
32
33
34 def main():
35   mat = [
36     [10, 3, 3, 1, 2, 1],
37     [1, 14, 1, 3, 5, 1],
38     [2, 1, 7, 1, 3, 2],
39     [1, 2, 2, 12, 3, 1],
40     [1, 4, 2, 2, 9, 1],
41     [2, 1, 2, 5, 4, 8],
42   ]
43   vec = [4, 1, 4, 6, 3, 6]
44
45   print(richardson(mat, vec, 16)[0])
```

```
46      print(linalg.solve(mat, vec))
47
48
49  if __name__ == '__main__':
50      main()
51
```

Листинг 1: Компьютерная реализация алгоритма

Вывод

В данной лабораторной работе был реализован итерационный метод Ричардсона.