



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 №5 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	2
Вывод	4

Введение

В данной лабораторной работе необходимо найти решение уравнения с помощью QR-разложения матрицы

Постановка задачи

Дана матрица A , необходимо ее разложить на $A = QR$ и решить СЛАУ $Ax = B$.

Алгоритм работы

1. С помощью QR-разложения мы можем представить матрицу A в виде $Q \cdot R$, где Q – ортогональная матрица ($QQ^T = I, Q^T = Q^{-1}$), а R – верхнеугольная матрица.
2. Чтобы найти матрицу Q , найдем матрицу $P = I - 2\frac{p, x}{p, p}p$, где p является вектором нормали.
3. Матрицу R несложно вычислить, зная $A = QR$, следовательно матрица вычисляется по формуле $R = Q^T A$.
4. Решение СЛАУ будет следующим: $x = R^{-1}Q^T B$.

Приложения

```
1  import numpy as np
2  from lab2 import SLE
3
4
5  def QR(A):
6      def step(a):
7          v = a / (a[0] + np.copysign(np.linalg.norm(a), a[0]))
8          v[0] = 1
9          P = np.eye(a.shape[0])
10         P -= (2 / np.dot(v, v)) * np.dot(v[:, None], v[None, :])
```

```

11     return P
12
13     m, n = A.shape
14     Q = np.eye(m)
15     for i in range(n):
16         P = np.eye(m)
17         P[i:, i:] = step(A[i:, i])
18         Q = np.dot(Q, P)
19         A = np.dot(P, A)
20     return np.matrix(Q), np.matrix(A)
21
22
23 def solve(A, b):
24     Q, R = QR(A)
25     Q = Q.tolist()
26     sle = SLE(Q, b.tolist())
27     sle.solve()
28     y = sle.get_vec()
29     sle = SLE(R.tolist(), y)
30     sle.reverse()
31     x = sle.get_vec()
32     return x
33
34
35 def main():
36     mat = np.array((
37         [1, 2, 4],
38         [3, 3, 2],
39         [4, 1, 3]
40     ), dtype=float)
41
42     vec = np.array([4, 2, 3])
43
44     print(solve(mat, vec.copy()))
45
46     sle = SLE(mat.tolist(), vec.tolist())
47     print(sle.solve()[1])
48
49
50 if __name__ == '__main__':
51     main()
52
53

```

Листинг 1: Компьютерная реализация алгоритма

Вывод

В данной лабораторной работе было произведено вычисление решения СЛАУ с помощью QR-разложения.