Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Псковский государственный университет»**

Передовая инженерная школа гибридных технологий в станкостроении

Союзного государства

Отделение информационно-коммуникационных технологий

Отчет по лабораторной работе №7

«Определение собственных векторов и собственных значений матрицы»

Вариант №22

**Выполнил:** Иванов И.С.

группа 0482-06

**Проверил:** Андреев Д. А.

Псков

2024

1. **Вариант задания**

Вариант №20: найти наибольшее по модулю собственное значение матрицы с точностью .

1. **Используемые вычислительные формулы**

Произвольный вектор , .

, первое приближение собственного значения , где .

, второе приближение собственного значения .

Условие окончания вычислений: .

1. **Блок-схема алгоритма**

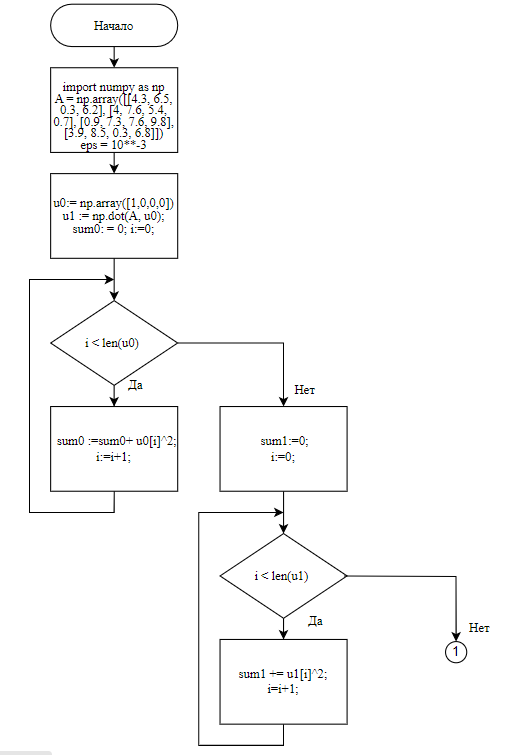


Рис. 1. Блок-схема алгоритма программы (часть 1)

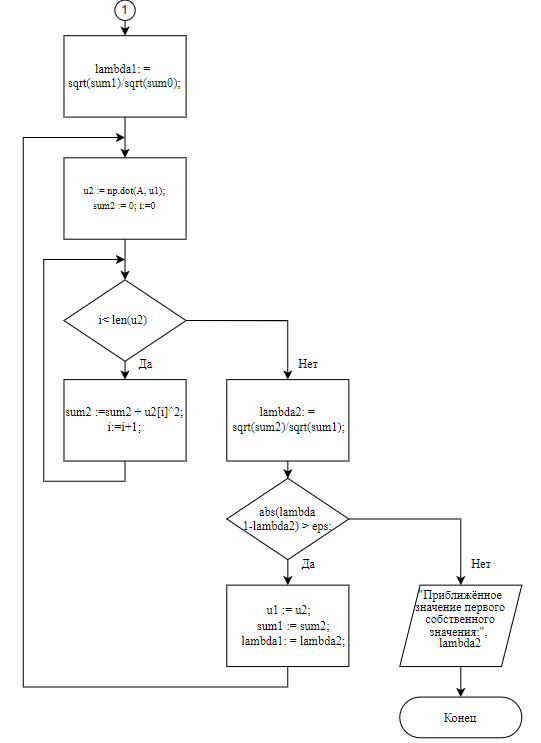


Рис. 2. Блок-схема алгоритма программы (часть 2)

1. **Текст программы на языке высокого уровня**

import numpy as np

import math

# Исходная матрица A

A = np.array([[4.3, 6.5, 0.3, 6.2],

[4, 7.6, 5.4, 0.7],

[0.9, 7.3, 7.6, 9.8],

[3.9, 8.5, 0.3, 6.8]])

# Параметры для итерации

eps = 10\*\*-3

u0 = np.array([1, 0, 0, 0]) # Начальное приближение для собственного вектора

# Первая итерация

u1 = np.dot(A, u0)

sum0 = np.sum(u0\*\*2)

sum1 = np.sum(u1\*\*2)

lambda1 = math.sqrt(sum1) / math.sqrt(sum0)

# Второе приближение

u2 = np.dot(A, u1)

sum2 = np.sum(u2\*\*2)

lambda2 = math.sqrt(sum2) / math.sqrt(sum1)

# Итерация для первого собственного значения

while abs(lambda1 - lambda2) > eps:

u1 = u2

sum1 = sum2

lambda1 = lambda2

u2 = np.dot(A, u1)

sum2 = np.sum(u2\*\*2)

lambda2 = math.sqrt(sum2) / math.sqrt(sum1)

# Нормализация собственного вектора

eigenvector1 = u2 / np.linalg.norm(u2)

print("Приближённое значение первого собственного значения:", lambda2)

print("Соответствующий собственный вектор:", eigenvector1)

# Для второго собственного значения

# Начальное приближение для второго собственного вектора (ортогонально первому)

u0\_second = np.array([0, 1, 0, 0]) # Начальное приближение, можно выбрать иначе по желанию

u1\_second = np.dot(A, u0\_second)

sum0\_second = np.sum(u0\_second\*\*2)

sum1\_second = np.sum(u1\_second\*\*2)

lambda1\_second = math.sqrt(sum1\_second) / math.sqrt(sum0\_second)

# Второе приближение

u2\_second = np.dot(A, u1\_second)

sum2\_second = np.sum(u2\_second\*\*2)

lambda2\_second = math.sqrt(sum2\_second) / math.sqrt(sum1\_second)

# Итерация для второго собственного значения

while abs(lambda1\_second - lambda2\_second) > eps:

u1\_second = u2\_second

sum1\_second = sum2\_second

lambda1\_second = lambda2\_second

u2\_second = np.dot(A, u1\_second)

sum2\_second = np.sum(u2\_second\*\*2)

lambda2\_second = math.sqrt(sum2\_second) / math.sqrt(sum1\_second)

# Нормализация второго собственного вектора

eigenvector2 = u2\_second / np.linalg.norm(u2\_second)

print("Приближённое значение второго собственного значения:", lambda2\_second)

print("Соответствующий собственный вектор:", eigenvector2)

# Проверка наличия третьего собственного значения

u0\_third = np.array([0, 0, 1, 0]) # Начальное приближение для третьего собственного вектора

u1\_third = np.dot(A, u0\_third)

sum0\_third = np.sum(u0\_third\*\*2)

sum1\_third = np.sum(u1\_third\*\*2)

lambda1\_third = math.sqrt(sum1\_third) / math.sqrt(sum0\_third)

# Второе приближение для третьего собственного значения

u2\_third = np.dot(A, u1\_third)

sum2\_third = np.sum(u2\_third\*\*2)

lambda2\_third = math.sqrt(sum2\_third) / math.sqrt(sum1\_third)

# Итерация для третьего собственного значения

while abs(lambda1\_third - lambda2\_third) > eps:

u1\_third = u2\_third

sum1\_third = sum2\_third

lambda1\_third = lambda2\_third

u2\_third = np.dot(A, u1\_third)

sum2\_third = np.sum(u2\_third\*\*2)

lambda2\_third = math.sqrt(sum2\_third) / math.sqrt(sum1\_third)

# Нормализация третьего собственного вектора

eigenvector3 = u2\_third / np.linalg.norm(u2\_third)

# Проверка возможности вывода третьего собственного значения

if np.abs(lambda2\_third - lambda2\_second) < eps: # Если значения близки, можно считать второе значение конечным

print("Третье собственное значение не найдено.")

else:

print("Приближённое значение третьего собственного значения:", lambda2\_third)

Рис. 3. Текст программы

1. **Результат**

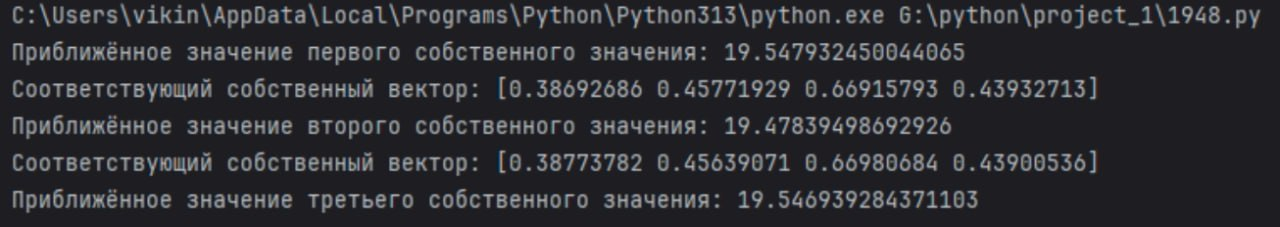


Рис. 4. Результат работы программы

1. **Вывод**

В этой лабораторной работе было вычислено наибольшее по модулю собственное значение заданной матрицы с заданной точностью. Был выбран произвольный вектор, который умножался на заданную матрицу. Затем последовательно происходило вычисление приближённых значений наибольшего по модулю собственного значения матрицы и приближённых значений собственного вектора.