**Министерство образования Республики Беларусь**

**Белорусский государственный университет**

Факультет прикладной математики и информатики

Лабораторная работа №8

По курсу «Вычислительные методы алгебры»

**Степенной метод решения частичной проблемы собственных значений.**

Вариант №5

Работу выполнил:

студент 3 курса 7 группы

**Шатерник Артём**

Преподаватель:

**Будник А. М**.

**Минск 2023**

1. **Постановка задачи.**

Дана матрица следующего вида

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0.5757 | -0.0758 | 0.0152 | 0.0303 | 0.1061 |  |
|  |  |  |  |  |  |
| 0.0788 | 0.9014 | 0.0000 | -0.0606 | 0.0606 |  |
|  |  |  |  |  |  |
| 0.0455 | 0.0000 | 0.7242 | -0.2121 | 0.1212 |  |
|  |  |  |  |  |  |
| -0.0909 | 0.1909 | 0.0000 | 0.7121 | -0.0303 |  |
|  |  |  |  |  |  |
| 0.3788 | 0.0000 | 0.1364 | 0.0152 | 0.8484 |  |

Требуется степенным методом найти максимальное собственное значение и соответствующий ему собственный вектор для матрицы Вычислить невязку оценить её значение. Точность

1. **Алгоритм решения.**

Метод является итерационным с заданной наперёд точностью.

Берётся некоторый начальный вектор Был взят .

Далее на каждой итерации вычисляется

То есть получается система вида

*, , … , …*

Так как матрица

является симметричной, то можем применять следующий метод для нахождения максимального собственного значения при каждой итерации

Критерием остановки итерационного процесса будет

В качестве собственного вектора, соответствующего данному собственному значению, берём

1. **Листинг программы.**

**import numpy as np  
import math  
  
size = 5  
a\_matrix = []  
with open('input.txt') as file:  
 i = 0  
 for line in file:  
 a\_matrix.append([float(x) for x in line.split(' ')])  
 i += 1  
a\_matrix = np.array(a\_matrix)  
epsilon = 1e-5  
# Симметричный вид  
a\_matrix = np.matmul(a\_matrix.T, a\_matrix)  
# Начальное приближение  
y\_old = [0 for i in range(size - 1)]  
y\_old.append(1)  
print("Начальное приближение:")  
print(y\_old)  
y\_new = np.matmul(a\_matrix, y\_old)  
lambda\_new = np.dot(y\_new, y\_old) / np.dot(y\_old, y\_old)  
n = 0  
while True:  
 n += 1  
 y\_old = y\_new  
 lambda\_old = lambda\_new  
 y\_new = np.matmul(a\_matrix, y\_new)  
 lambda\_new = np.dot(y\_new, y\_old) / np.dot(y\_old, y\_old)  
 if abs(lambda\_new - lambda\_old) <= epsilon:  
 break  
print("Число итераций: " + str(n))  
print('Максимальное собственное значение:')  
print(lambda\_new)  
print('Собственный вектор, соответствующий этому собственному значению:')  
print(y\_old)**

**# Невязка для собственного значения**

**# Из метода Данилевского**

**p = [3.1966884499999972, -3.7968475734971836, 2.0678062361750578, -0.5082483413019262, 0.044096040836178144]**

**sum = pow(lambda\_new, size)**

**for j in reversed(range(size)):**

**sum -= pow(lambda\_new, j) \* p[size - j - 1]**

**print(format(sum, '.4e'))**

**# Невязка для собственного значения и собственного вектора**

**res = np.matmul(a\_matrix, y\_old) - lambda\_new \* y\_old**

**print(res)**

**print('Норма невязки: ', end='')**

**print(format(np.linalg.norm(res, 2), '.4e'))**

1. **Результат и его анализ.**

Число итераций: 9

Максимальное собственное значение:

1.1447499595770831

Собственный вектор, соответствующий этому собственному значению:

[ 1.29976058 0.18863179 0.94458672 -0.44767659 1.85494189]

Невязка для максимального собственного значения. Собственный многочлен был взят из метода Данилевского

-6.5667e-07.

Невязка

[-0.00034888 -0.00244649 0.00111387 -0.00198575 -0.00055321]

Норма невязки: 3.4054e-03

Если вместо собственного значения, вычисленного степенным методом взять собственное значение с точность то получим

[-0.00035698 -0.00244767 0.00110798 -0.00198296 -0.00056478]

Норма невязки: 3.4055e-03

Экономичность:

Число операций при одной итерации порядка

Из-за использования скалярных произведений при вычислении собственных значений увеличилось число операций, но на порядок увеличилась скорость сходимости метода.

Точность:

Метод является итерационным и даёт наперёд заданную точность.

Из невязки видно, что собственное значение было вычислено с точностью даже выше заданной.

По невязкам видно, что и при собственном значении порядка подсчитанного степенным методом и при использовании собственного значения порядка точность вычисления собственного вектора получилась порядка , что ниже, чем наперёд заданная точность.