

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ     
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего профессионального образования

**ДАЛЬНЕВОСТОЧНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ**

**Институт математики и компьютерных технологий**

**Департамент математического и компьютерного моделирования**

**ОТЧЕТ**

**по лабораторной работе** **№ 8**

«Метод итераций для нахождения собственных чисел и векторов матрицы»

Выполнил: студент гр. Б9120-02.03.01сцт

Пограничный К. О.

Проверил: преподаватель

Кузнецов К. С.

Владивосток 2023

**Постановка задачи**

Дана симметричная положительно определенная матрица А размерности n. Требуется найти её собственные значения методом итераций и изучить, сколько требуется шагов методу итераций для каждого случая при разных

В качестве входных данных будем использовать точность ; начальный вектор , состоящий из ‘1’ и матрицы:

**Метод решения**

Метод итераций для нахождения собственных значений и соответствующих собственных векторов матрицы представляет собой численный алгоритм, основанный на итеративном процессе.

Начиная с некоторых начальных приближений собственных значений и векторов, на каждой итерации выполняются шаги, включающие умножение матрицы на текущий приближенный вектор и его нормировку, а также вычисление собственного значения на основе скалярного произведения полученного вектора с матрицей.

**Алгоритм решения**

Метод итераций можно описать следующей схемой:

1. Выбрать произвольное начальное приближение собственного вектора и задать точность решения,
2. С
3. Вычислить
4. Найти размерность матрицы
5. Повторять шаги 2-3 до тех пор, пока не будет выполняться условие где, задано на первом шаге.

**Подробный анализ кода**

Для реализации алгоритмов решения используется язык программирования python и библиотека numpy.

np.linalg.norm, использующаяся для нахождения нормы разности – встроенная функция numpy, находящая Евклидову норму без указания параметра

**Результаты тестов**

После проведения численного эксперимента с заданными матрицами и точностью, были получены следующие данные:

1. Рассмотрим матрицу :

Вычисленный встроенной функцией собственный вектор: [10.229815083 5.491796623 2.278388294]

Значения, вычисленные методом вращений:

|  |  |  |  |
| --- | --- | --- | --- |
|  | max собственное значение | Собственный вектор | Кол-во шагов |
|  | 10.331140350877192 | [3.354296519 3.972832534 8.803393258] | 4 |
|  | 10.230497724600053 | [3.19032803 3.884949121 8.909438173] | 12 |
|  | 10.229815811475268 | [3.189170832 3.884320512 8.910126834] | 23 |
|  | 10.229815083731111 | [3.189169596 3.884319841 8.910127569] | 34 |

1. Рассмотрим матрицу :

Вычисленный встроенной функцией собственный вектор: [17.059458597 -2.939416225 3.263086073 8.214584626 10.402286929]

Значения, вычисленные методом вращений:

|  |  |  |  |
| --- | --- | --- | --- |
|  | max собственное значение | Собственный вектор | Кол-во шагов |
|  | 17.16936870914093 | [8.023672683 0.767367031 8.452905348 11.052341517 5.68105339] | 5 |
|  | 17.060544233775637 | [8.013733808 0.650841904 8.534543352 11.147022206 5.411750619] | 14 |
|  | 17.05945964302285 | [8.01376126 0.649500478 8.535385046 11.148015759 5.408498309] | 28 |
|  | 17.05945859756281 | [8.013761311 0.649499165 8.535385863 11.148016727 5.408495107] | 42 |

1. Рассмотрим матрицу :

Вычисленный встроенной функцией собственный вектор: [16.950852322 -0.740019009 9. 3.325873293 4.1189271 6.842371148 6.501995145]

Значения, вычисленные методом вращений:

|  |  |  |  |
| --- | --- | --- | --- |
|  | max собственное значение | Собственный вектор | Кол-во шагов |
|  | 16.99176180255286 | [8.523982734 1.71468926 3.992393597 9.781220265 4.394464258 2.854818361  8.523982734] | 5 |
|  | 16.95127578620507 | [8.520173053 1.704464335 3.998874957 9.791599085 4.386916171 2.850973661  8.520173053] | 10 |
|  | 16.950852587455582 | [8.520125877 1.704360863 3.998940835 9.791716621 4.386838816 2.850940478  8.520125877] | 18 |
|  | 16.950852322746115 | [8.520125842 1.704360797 3.998940876 9.791716707 4.386838762 2.850940456  8.520125842] | 25 |

**Вывод**

В процессе работы было установлено, что метод итераций обладает высокой гибкостью и точностью. Задавая требуемую точность ε, можно контролировать точность полученных приближенных решений. Однако стоит отметить, что с увеличением требуемой точности увеличивается количество шагов итераций.

При увеличении размерности матрицы количество шагов особо не увеличивается, но растет количество математических операций, связанных с выполнением матричных умножений. Это может привести к увеличению времени выполнения метода. Следовательно, увеличение размерности матрицы может значительно влиять на производительность метода итераций.

В заключение, метод итераций для нахождения собственного вектора и значения матрицы является мощным инструментом для численного решения задачи нахождения собственных значений.

**Приложение**

Листинг программы:

import numpy as np  
import warnings  
  
  
warnings.simplefilter(action="ignore", category=RuntimeWarning)  
np.set\_printoptions(precision=9)  
  
  
def iterations\_method(A: np.array, x\_0: np.array, accuracy: float):  
 x\_new = A.dot(x\_0)  
  
 eigval\_old = np.inf  
 eigval\_new = np.inf  
 steps\_num = 0  
 while (True):  
 steps\_num += 1  
 x\_old = x\_new / np.linalg.norm(x\_new)  
 x\_new = A.dot(x\_old)  
  
 eigval\_old = eigval\_new  
 eigval\_new = max(x\_new / x\_old)  
  
 if (abs(eigval\_new - eigval\_old) < accuracy):  
 break  
  
 return eigval\_new, x\_new, steps\_num  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 A1 = np.array([  
 [5, 2, 1],  
 [2, 4, 2],  
 [1, 2, 9]  
 ])  
 A2 = np.array([  
 [10, 1, 2, 3, 1],  
 [1, 2, -3, 1, 3],  
 [2, -3, 5, 7, 2],  
 [3, 1, 7, 9, 1],  
 [1, 3, 2, 1, 10]  
 ])  
 A3 = np.array([  
 [10, 1, 2, 3, 2, 1, 1],  
 [1, 4, -3, 1, 1, 1, 1],  
 [2, -3, 3, 2, 1, 1, 2],  
 [3, 1, 2, 10, 1, 1, 3],  
 [2, 1, 1, 1, 5, 1, 2],  
 [1, 1, 1, 1, 1, 4, 1],  
 [1, 1, 2, 3, 2, 1, 10]  
 ])  
  
 A\_list = [A1, A2, A3]  
 acc\_list = [1e-1, 1e-3, 1e-6, 1e-9]  
  
 for A in A\_list:  
 print("Matrix: \n{}\n\n".format(A))  
 print("calculated by the built-in function eigvals: " + str(np.linalg.eigvals(A)) + "\n\n")  
  
 for accuracy in acc\_list:  
 print("accuracy: {}\n".format(accuracy))  
 n = A.shape[0]  
 eigval, eigvect, steps\_num = iterations\_method(A, np.zeros(n) + 1, accuracy)  
 print("found eigval: " + str(eigval))  
 print("found eigvect: " + str(eigvect))  
 print("steps num = " + str(steps\_num) + "\n")