БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Отчет по лабораторной работе №3

по теме:«Итерационный степенной метод»

Выполнил:
Гаргома А. О.

Преподаватель:
Горбачева Ю.Н.

Минск 2019

Оглавление

1 Постановка задачи

С помощью итерационного степенного метода найти с точностью 10^6 наибольшее по модулю собственное значение и соответствующий ему собственный вектор симметричной матрицы. Вычислительный процесс проводить с нормировкой векторов итерационной последовательности.

2 Теоретические сведения

Итерационный степенной метод (называемый также степенным методом) предназначен для нахождения одного или нескольких собственных значений и соответствующих собственных векторов. Пусть A – вещественная матрица порядка n. Любая матрица преобразованием подобия $S^{-1}AS$ с подходящей матрицей подобия S может быть приведена к жордановой (нормальной) форме: на главной диагонали – собственные значения, на наддиагонали – нули и/или единицы. Матрица заведомо диагонализируема в двух важных частных случаях: если она симметричная или если ее собственные значения различны. Диагонализируемая матрица имеет ровно n линейно независимых собственных векторов.

Лемма 1. *Имеют место соотношения*

$$\frac{y_i^{k+1}}{y_i^k} = \lambda_1 + O\left(\left|\frac{\lambda_2}{\lambda_1}\right|\right),\ i = 1, 2, \ldots, n,$$

$$y_i^k = \alpha_1 \lambda_1^k x_{1i} \left(1 + O\left(\left|\frac{\lambda_2}{\lambda_1}\right|\right)\right),\ i = 1, 2, \ldots, n.$$

Следствие 1. *При достаточно больших значениях k выполняются приближенные равенства*

$$\lambda_1 \approx \frac{y_i^{k+1}}{y_i^k}, i = 1, 2, \ldots, n.$$

Лемма 2. *Если матрица A симметричная, то имеет место соотношение*

$$\frac{(y^{k+1}, y^k)}{(y^k,\ y^k)} = \lambda_1 + O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right)$$

Следствие 2. *При достаточно больших значениях k выполняется приближенное равенство*

$$\lambda_1 \approx \frac{(y^{k+1},\ y^k)}{(y^k,\ y^k)}$$

причем сходимость отношения $\frac{(y^{k+1},\ y^k)}{(y^k,\ y^k)}$ более быстрая, чем сходимость отношений $\frac{y_i^{k+1}}{y_i^k}$ (погрешность возводится в квадрат).

3 Листинг программы

```
# Вариант 2
import numpy as np
import tabulate as tb
import sympy as sp

# Степенной метод со скоростью сходимости (lambda_2/
    lambda_1)^k
def power_iteration1(B,u_k):
    A = np.copy(B)
    v_k = A@u_k
    u_k1 = (v_k)/np.linalg.norm(v_k,np.inf) #
        нормирование
    l1 = v_k/u_k
    u_k = u_k1
    v_k = A@u_k
```

```
    u_k1 = (v_k)/np.linalg.norm(v_k,ord=np.inf)
    l2 = v_k/u_k
    count = 2
    # Итерационный процесс
    while(np.linalg.norm(l2-l1,ord=np.inf)>=accuracy):
        u_k = u_k1
        v_k = A @ u_k
        u_k1 = (v_k) / np.linalg.norm(v_k, ord=np.inf)
        l1 = l2
        l2 = v_k/u_k
        count+=1
    return sum(l2)/len(l2),u_k1, count

# Степенной метод со скоростью сходимости (lambda_2/
    lambda_1)^2k
def power_iteration2(B,u_k):
    A = np.copy(B)
    v_k = A @ u_k
    u_k1 = (v_k) / np.linalg.norm(v_k, ord=np.inf)
    l1 = (v_k @ u_k)/(u_k@u_k)
    u_k = u_k1
    v_k = A @ u_k
    u_k1 = (v_k) / np.linalg.norm(v_k, ord=np.inf)
    l2 = (v_k @ u_k)/(u_k@u_k)
    count = 2
    # Итерационный процесс
    while (abs(l2 - l1) >= accuracy):
        u_k = u_k1
        v_k = A @ u_k
        u_k1 = (v_k) / np.linalg.norm(v_k, ord=np.inf)
        l1 = l2
        l2 = (v_k @ u_k)/(u_k@u_k)
        count += 1
    return l2,u_k1, count


B = np.array([[1.342, 0.432, 0.599, 0.202, 0.603,
    0.202],
              [0.432, 1.342, 0.256, 0.599, 0.204,
                0.304],
              [0.599, 0.256, 1.342, 0.532, 0.101,
```

```
                 0.506],
               [0.202, 0.599, 0.532, 1.342, 0.106,
                 0.311],
               [0.603, 0.204, 0.101, 0.106, 1.342,
                 0.102],
               [0.202, 0.304, 0.506, 0.311, 0.102,
                 1.342]])
C = np.array([[0.05, 0, 0, 0, 0, 0],
               [0, 0.03, 0, 0, 0, 0],
               [0, 0, 0.02, 0, 0, 0],
               [0, 0, 0, 0.04, 0, 0],
               [0, 0, 0, 0, 0.06, 0],
               [0, 0, 0, 0, 0, 0.07]])

k = 2
accuracy = .1e-5
A = B + k*C
print('Матрица A: \n',tb.tabulate(A))
u_k = np.asarray([1,1,1,1,1,1])
answer1 = power_iteration1(A,u_k)
u_k = np.asarray([1,1,1,1,1,1])
answer2 = power_iteration2(A,u_k)
print('Начальный вектор: ', u_k)
print(tb.tabulate([answer1[:2],answer2[:2]],headers=['
   Макс. собств. зн.',
"Собственный вектор"],floatfmt=".10f"))
print('Количество итераций для достижения заданной 
   точности в двух способах'
      ' соответсвенно: ',
      answer1[2],' ',answer2[2])
```

4 Результаты эксперимента

```
Матрица A:
-----  -----  -----  -----  -----  -----
1.442  0.432  0.599  0.202  0.603  0.202
0.432  1.402  0.256  0.599  0.204  0.304
0.599  0.256  1.382  0.532  0.101  0.506
0.202  0.599  0.532  1.422  0.106  0.311
0.603  0.204  0.101  0.106  1.462  0.102
0.202  0.304  0.506  0.311  0.102  1.482
```

```
-----  -----  -----  -----  -----  -----
Начальный вектор:  [1 1 1 1 1 1]
Макс. собств. зн.  Собственный вектор
------------------  ------------------------------------------------------
3.1627520967  [0.99655425 0.9103678  1. 0.91428199 0.62643134 0.79267685]
3.1627518951  [0.99686455 0.91039785 1. 0.91418272 0.62683012 0.79259295]
Количество итераций для достижения заданной точности
 в двух способах соответственно:  21   9
```

5 Выводы

Степенной метод или метод степенных итераций — итерационный алгоритм поиска собственного значения с максимальной абсолютной величиной и одного из соответствующих собственных векторов для произвольной матрицы.

Из результатов эксперимента можно заключить, что алгоритм степенного метода, учитывающий симметричность матрицы, сходится в 2 раза быстрее