

Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

ОТЧЕТ

ПО ПРЕДМЕТУ “ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ АЛГЕБРЫ”

студентки 2 курса 2 группы

Курец Любови Олеговны

Преподаватель

Горбачева Юлия Николаевна

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.

Краткие теоретические сведения:

Суть метода заключается в последовательном приближении y^k к собственному вектору соответствующему максимальному собственному значению λ . За λ^k берётся отношение соответствующий произвольных координат векторов y^{k+1} и y^k . Итерационный процесс останавливается, когда $|\lambda^{k+1} - \lambda^k| \leq \varepsilon$.

Возьмём начальное приближение $y^0 = (1, 1, \dots, 1)$.

При вычислениях последовательности на компьютере координаты вектора y^k могут сильно расти, что может привести к переполнению, а могут сильно убывать, что может привести к машинному нулю. Поэтому на практике требуется производить нормировку. Для организации нормированных вычислений введем в рассмотрение следующие две одновременно вычисляемые последовательности:

$$u^0 = y^0, \\ v^k = A u^{k-1}, \quad u^k = \frac{v^k}{\|v^k\|}, \\ k = 1, 2, \dots$$

Если матрица A симметричная, то имеет место соотношение:

$$\lambda_1 \approx \frac{(v^{k+1}, u^k)}{(u^k, u^k)}.$$

Листинг программы:

```
import numpy as np
import tabulate as tb
import sympy as sp

#метод простых итераций
def powerIteration(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

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

#метод скалярных произведения
def powerIteration_scalar(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

    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 = 3
accuracy = .1e-5
A = B + k*C
print('Матрица A:\n',tb.tabulate(A))
u_k = np.asarray([1,1,1,1,1,1])
result_1 = powerIteration(A,u_k)
u_k = np.asarray([1,1,1,1,1,1])
result_2 = powerIteration_scalar(A,u_k)
print('\nY_0: ', u_k)
print('\nМаксимальное собственное значение:')
print(result_1[0])
print('Собственный вектор')

```

```

for i in result_1[1:2]:
    print(i,end = ' ')
print('\nКол-во итераций:',result_1[2])
print('\n\nМаксимальное собственное значение:')
print(result_2[0])
print('Собственный вектор')
for i in result_2[1:2]:
    print(i,end = ' ')
print('\nКол-во итераций:',result_2[2],'\n')

```

Результаты вычислительного эксперимента:

```

Матрица A:
-----
1.492  0.432  0.599  0.202  0.603  0.202
0.432  1.432  0.256  0.599  0.204  0.304
0.599  0.256  1.402  0.532  0.101  0.506
0.202  0.599  0.532  1.462  0.106  0.311
0.603  0.204  0.101  0.106  1.522  0.102
0.202  0.304  0.506  0.311  0.102  1.552
-----

Y_0:  [1 1 1 1 1 1]

Максимальное собственное значение:
3.2046629910085307
Собственный вектор
[1.          0.90522478 0.99054866 0.91123399 0.63361927 0.8026008 ]
Кол-во итераций: 21

Максимальное собственное значение:
3.2046627557074174
Собственный вектор
[1.          0.90494773 0.99020834 0.91081323 0.63384982 0.80222889]
Кол-во итераций: 9

Press any key to continue . . .

```

Выводы:

С помощью степенного метода мы нашли максимальное по модулю собственное значение и соответствующий ему собственный с точностью порядка 10^{-16} за 21 итерацию для эпсилона порядка 10^{-6} , и 9 итераций, если считать методом скалярных произведений. Чтобы СМ сходиллся необходимо и достаточно, чтобы у матрицы были доминирующее собственное значение.

