**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Факультет прикладной математики и информатики

Лабораторная работа №6 по курсу “ВМА”

“Метод Вращения”

Вариант №3

Выполнил: Ёда Никита

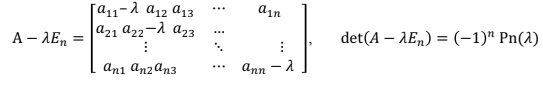
3 курс, 6(а) группа

Преподаватель: Будник А.М.

2023

**Постановка задачи**

Необходимо найти максимальное собственное значение и соответствующий ему собственный вектор матрицы А.

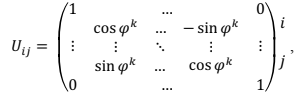


С помощью метода вращений найти спектр матрицы A. Вычислить собственный вектор, соответствующий максимальному по модулю собственному значению.

**Алгоритм решения**

Метод вращений является итерационным методом решения полной проблемы собственных значений. Суть метода заключается в привидении матрицы А к диагональному виду с помощью подобных преобразований: (, где ― ортогональная матрица, Λ ― диагональная матрица, на диагонали которой стоят собственные значения. В силу ортогональности U, получаем Λ.

На каждом шаге итерации строиться матрица



где





а i, j получаем как индексы максимального недиагонального элемента матрицы Последовательно выполняя , придем к диагональной матрице. Тогда - координатные столбцы матрицы U образуют соответственно координатные столбцы собственных векторов соответствующих собственным значениям, стоящим а диагональ матрицы Λ. Итерационный процесс заканчивается, когда ,

**Листинг программы**

a = np.array(A)

n = 5

print("Матрица A:\n\t", A)

At = a.transpose()

a = np.dot(At, a)

print("\nСимметрическая матрица A\*A^T:\n\t", a)

E = np.identity(n)

U = np.identity(n)

eps = 10 \*\* (-5)

k = 0

ak = a

while (True):

    L = np.tril(ak)

    temp = np.absolute(ak - L)

    sigma = sum([abs(el) \*\* 2 for el in temp])

    if np.all(sigma <= eps):

        break

    i, j = np.unravel\_index(temp.argmax(), temp.shape)

    alpha = math.atan(2 \* ak[i][j] / (ak[i][i] - ak[j][j])) / 2

    uk = np.identity(n)

    uk[i][i] = math.cos(alpha)

    uk[i][j] = -math.sin(alpha)

    uk[j][j] = math.cos(alpha)

    uk[j][i] = math.sin(alpha)

    ak = np.dot(np.dot(uk.transpose(), ak), uk)

    U = np.dot(U, uk)

    k += 1

print("\nМатрица Λ=U^T\*AU:\n\t", ak)

p = [4.58801522, -7.82119475, 6.11344651, -2.15665219, 0.2685558]

print("\nКоэффициенты собственного многочлена P(lambda):\n\t", p)

maxLambda = max(ak.diagonal())

print("\nmax(lambda):\n\t", maxLambda)

print("\nМатрица U:\n\t", U)

x = U.transpose()[ak.diagonal().argmax()]

x /= max(x)

print("\nСобственный вектор матрицы А — x(max(lambda))∶\n\t", x)

print("\nКол-во:\n\t", k)

print("\nEpsilon:\n\t", eps)

r = np.dot(a, x) - maxLambda \* x

print("\nВектор невязки r:\n\t", r)

p.insert(0, -1)

r1 = sum(-(maxLambda \*\* (n - i)) \* p[i] for i in range(n + 1))

print("\nНевязка Pn(lambda^k):\n\t", r1)

**Вывод**

Симметрическая матрица A\*A^T:

[[ 0.22060583 0.00450325 0.04193006 -0.02474925 0.17753974]

[ 0.00450325 0.38363426 -0.00051816 0.03470411 0.01831009]

[ 0.04193006 -0.00051816 0.24387704 -0.06780758 0.09207522]

[-0.02474925 0.03470411 -0.06780758 0.24429211 -0.0155092 ]

[ 0.17753974 0.01831009 0.09207522 -0.0155092 0.33678766]]

Матрица Λ=U^T\*AU:

[[ 8.56846578e-02 -2.44618572e-03 2.01041478e-04 9.20586171e-06

9.11515803e-19]

[-2.44618572e-03 3.93983409e-01 -1.12545345e-04 -4.43614358e-04

3.20385380e-05]

[ 2.01041478e-04 -1.12545345e-04 1.70119009e-01 -9.24905107e-19

6.93312934e-04]

[ 9.20586171e-06 -4.43614358e-04 -2.71994810e-17 2.65832218e-01

-2.26087158e-03]

[-5.69696415e-18 3.20385380e-05 6.93312934e-04 -2.26087158e-03

5.13577606e-01]]

Коэффициенты собственного многочлена P(lambda):

[1.4291969, -0.758323068, 0.184357781, -0.0202232134, 0.000783880398]

max(lambda):

0.5135776057514412

Матрица U:

[[ 7.52906585e-01 -2.08595498e-03 -3.21116187e-01 2.38681228e-01

5.22535155e-01]

[ 9.35055163e-04 9.55441918e-01 -8.99555679e-02 -2.71920836e-01

7.13927269e-02]

[ 2.17530700e-01 -1.08431087e-01 7.27222062e-01 -5.23151613e-01

3.71999403e-01]

[ 1.47875384e-01 2.73227693e-01 5.99892033e-01 7.17211543e-01

-1.70928784e-01]

[-6.03277768e-01 2.67528862e-02 8.36726264e-03 2.84623123e-01

7.44486323e-01]]

Собственный вектор матрицы А — x(max(lambda))∶

[ 0.70187341 0.09589528 0.49967258 -0.22959291 1. ]

Кол-во:

11

Epsilon:

1e-05

Вектор невязки r:

[-0.00102397 0.00078312 0.00226129 -0.00160763 -0.00085541]

Невязка Pn(lambda^k):

-9.59026108145189e-08

**Анализ**

С помощью метода вращений мы нашли максимальные все собственные значения за 11 итераций для эпсилона порядка 10-5. Невязка собственного многочлена также довльно близка к нулю, что означает, что собственное значение также найдено правильно. Собственное значение и собственный вектор также совпадает с полученым ранее методом Данилевского. Сравнивая со степенным методом имеем гораздо более высокую скорость сходимости, при незначительном ухудшении точности. Построенный итерационный процесс является сходящимся, так как t(Ak)->0, при k-> бесконечности.