**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Факультет прикладной математики и информатики

Лабораторная работа №7 по курсу “ВМА”

“Метод Крылова”

Вариант №3

Выполнил: Ёда Никита

3 курс, 6(а) группа

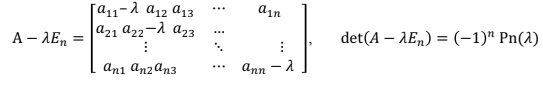
Преподаватель: Будник А.М.

2023

**Постановка задачи**

Необходимо найти собственные значения и собственные векторы

матрицы А:



Для этого требуется:

1. Построить - собственный (характеристический) многочлен матрицы А.

2. Решить уравнение:

3. Найти собственные векторы: , i = 1,n

**Алгоритм решения**

Метод Крылова является прямым методом решения полной задачи собственных значений. Согласно теореме Гамильтона-Кэли, матрица А удовлетворяет своему характеристическому многочлену, поэтому:

= 0.

Возьмём произвольный, ненулевой вектор:

согласованный с исходной матрицей и рекуррентно вычислим:

будет являться линейно-независимой комбинацией векторов

Покоординатно расписывая это равенство, придём к системе из n линейных уравнений от n неизвестных:

В матричном виде:

Решая её найдём коэффиценты характеристического многочлена

Из уравнения = 0 находим алее найдём собственный вектор x соответствующий собственному значению . Для этого распишем x в базисе:

Найдём по формулам:

**Листинг программы**

import numpy as np

from sympy import Symbol, solve

A = [[0.3857, -0.0508, 0.0102, 0.0203, 0.0711],

    [0.0528, 0.6039, 0.0000, -0.0406, 0.0406],

    [0.0305, 0.0000, 0.4852, -0.1421, 0.0812],

    [-0.0609, 0.1279, 0.0000, 0.4711, -0.0203],

    [0.2538, 0.0000, 0.0914, 0.0102, 0.5684]]

n = 5

a = np.array(A)

At = a.transpose()

print("Симметрическая A\*A^T:\n\t", At)

a = np.dot(At, a)

c = []

c.append([1, 0, 0, 0, 0])

for i in range(1, n + 1):

    c.append(np.dot(a, c[i - 1]))

print("\nВекторы C^i:")

for el in c:

    print("\t", el)

C = np.array(c)

cn = c.pop()

c = np.array(c).transpose()

for i in range(n):

    c[i] = list(reversed(c[i]))

p = np.linalg.solve(c, cn)

print("\nКоэффициенты собственного многочлена P(lambda):\n\tБерём из прошлого метода")

x = Symbol('x')

Lambda = solve(x\*\*5 - p[0] \* x\*\*4 - p[1] \* x\*\*3 - p[2] \* x\*\*2 - p[3] \* x - p[4], x)

print("\nСобственные значения lambda:\n\t", Lambda)

l = max(Lambda)

print("\nМаксимальное собственное lambdaMax:\n\t", l)

b = np.ones(n)

for i in range(1, n):

    b[i] = b[i - 1] \* l - p[i - 1]

print("\nКоэффиценты B^i:\n\t", b)

x = np.sum([b[i] \* C[n - i - 1] for i in range(n)], axis=0)

print("\nСобственный вектор матрицы A - x(lambdaMax):\n\t", x)

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

print("\nВектор невязки r:\n\t", r)

rnorm = np.linalg.norm(r, 1)

print("\nНорма невязки ||r||:\n\t", rnorm)

**Вывод**

Симметрическая A\*A^T:

[[ 0.3857 0.0528 0.0305 -0.0609 0.2538]

[-0.0508 0.6039 0. 0.1279 0. ]

[ 0.0102 0. 0.4852 0. 0.0914]

[ 0.0203 -0.0406 -0.1421 0.4711 0.0102]

[ 0.0711 0.0406 0.0812 -0.0203 0.5684]]

Векторы C^i:

[1, 0, 0, 0, 0]

[ 0.22060583 0.00450325 0.04193006 -0.02474925 0.17753974]

[ 0.08257823 0.00509119 0.03749866 -0.01694627 0.10328651]

[ 0.03856935 0.00360867 0.02326415 -0.01015149 0.05325528]

[ 0.01920651 0.00216885 0.01288078 -0.00571269 0.02714888]

[ 0.00974832 0.00121071 0.00683263 -0.00309011 0.01386764]

Коэффициенты собственного многочлена P(lambda):

Берём из прошлого метода

Собственные значения lambda:

[0.0856647753716902, 0.170118025536942, 0.265810054944383, 0.394004397801629, 0.513599646345256]

Максимальное собственное lambdaMax:

0.513599646345256

Коэффиценты B^i:

[ 1. -0.91559725 0.28807264 -0.03640377 0.00152625]

Собственный вектор матрицы A - x(lambdaMax):

[ 0.00117641 0.00016746 0.00085611 -0.0003988 0.0016794 ]

Вектор невязки r:

[5.56195714485064e-17 4.20128341838133e-19 -9.21571846612679e-19 7.58941520739853e-19 4.87890977618477e-18]

Норма невязки ||r||:

6.25991229338818e-17

**Анализ**

С помощью метода Крылова мы нашли все собственные значения и по ним можем построить полный спектр. Для найденного собственного вектора x матрицы А соответствующего максимальному собственному значению норма вектора невязки:

r = Ax-x

получилась порядка 10-17, что говорит о том, что собственный вектор найден с достаточной точностью. Сравнивая с методом Данилевского, метод Крылова является более точным (в методе Данилевского невязка порядка 10-11), однако и более трудоёмким.