**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

Лабораторная работа №5

Вариант 5

«Метод Данилевского и итерационный степенной метод»

Выполнил: Благодарный Артём Андреевич,

студент 3 курса, 3 группы

Дисциплина: «Численные методы»

Преподаватель: Будник А.М.

Минск, 2024

A math equations and formulas

Description automatically generated with medium confidence1. Найти с помощью метода Данилевского форму Фробениуса, характеристический многочлен, r1 = p1 – SpA и r2=p5 – detA.

2. С помощью степенного метода найти максимальное собственное значение, определить количество итераций для eps=1е-5.

A math equations and formulas

Description automatically generated with medium confidence3. С помощью метода Данилевского найти собственный вектор, соответствующий максимальному собственному значению, найти невязку собственного значения и собственного вектора.

**Алгоритм степенного метода**

**A white background with black text

Description automatically generatedA white sheet with black text

Description automatically generated**

**Листинг**

import numpy as np

from tabulate import tabulate

from sympy import Symbol, solve

A = np.array([[0.8894, 0.0000, -0.2323, 0.1634, 0.2723],

[-0.0545, 0.5808, 0.0000, -0.1107, 0.0363],

[0.0182, -0.1634, 1.0527, 0.0200, 0.0635],

[0.0545, 0.0000, -0.1325, 1.0527, 0.0000],

[0.0363, -0.0545, 0.2632, -0.0218, 0.7623]])

A = A.T @ A

def pretty\_print(X, p, r1, r2, eigenvalue, eigenvector, r\_eigenvalue, r\_eigenvector, c=None):

p \*= -1

x = [f"x^{i}" if i != 1 else "x" for i in range(5, 0, -1)]

polynom = "p(x)="

for i, j in zip(x, p):

polynom += i

polynom += f" - {-round(j, 3)}" if np.sign(j) == -1 else f" + {round(j, 3)}"

print(f'1)Frobenius normal form:\n{tabulate(X, tablefmt="grid", floatfmt=".3f")}\n',\

f'2)Characteristic polynomial:\n{polynom}\n',\

f'3)r1 = p1 - SpA = {r1:.3e}\n',\

f'4)r2 = p5 - detA = {r2:.3e}\n',\

f'5)max eigenvalue: {eigenvalue}\n',\

f'6)eigenvector:\n{tabulate([eigenvector], tablefmt="grid", floatfmt=".5f")}\n',\

f'7)r of eigenvalue: {r\_eigenvalue}\n',\

f'8)r of eigenvector:\n{tabulate([r\_eigenvector], tablefmt="grid", floatfmt=".3e")}\n',\

"" if c is None else f'9)count of iterations: {c}')

def danilevsky\_method(A: np.ndarray):

X = A.copy()

n = X.shape[0]

s = np.eye(n)

n -= 1

for i in np.arange(n):

ones\_left = np.eye(n+1)

ones\_left[n-1-i] = X[n-i]

ones\_right = ones\_left.copy()

ones\_right[n-1-i] /= -X[n-i, n-1-i]

ones\_right[n-1-i, n-1-i] = 1 / X[n-i, n-1-i]

X = ones\_left @ X @ ones\_right

s = s @ ones\_right

p = X[0]

r1 = p[0] - np.trace(A)

detA = np.linalg.det(A)

r2 = p[n] - detA

return X, r1, r2, s, p

def power\_iteration(A: np.ndarray, num\_iter: int=1000, tol: float=1e-5):

n = A.shape[0]

u = np.zeros(n)

u[0] = 1

prev\_eigenvalue = 0

c = 0

for k in range(1, num\_iter + 1):

c += 1

v = A @ u

v\_norm = np.linalg.norm(v, np.inf)

u = v / v\_norm

eigenvalue = v\_norm

if abs(eigenvalue - prev\_eigenvalue) < tol:

break

prev\_eigenvalue = eigenvalue

return eigenvalue, c

def danilevsky\_power\_method(A: np.ndarray, type\_eigenvector='danilevsky'):

X = A.copy()

n = X.shape[0]

X, r1, r2, s, p = danilevsky\_method(X)

eigenvalue, u, c = power\_iteration(X)

r\_eigenvalue = np.sum([p[n-1-i] \* (eigenvalue \*\* i) for i in range(n)]) - eigenvalue \*\* (n)

if type\_eigenvector == 'danilevsky':

y = np.array([eigenvalue \*\* i for i in np.arange(n-1, -1, -1)])

eigenvector = s @ y

r\_eigenvector = X @ eigenvector - eigenvalue \* eigenvector

else:

r\_eigenvector = X @ u - eigenvalue \* u

eigenvector = u

return X, p, r1, r2, c, eigenvalue, eigenvector, r\_eigenvalue, r\_eigenvector

X, p, r1, r2, c, eigenvalue, eigenvector, r\_eigenvalue, r\_eigenvector = danilevsky\_power\_method(A)

pretty\_print(X, p, r1, r2, eigenvalue, eigenvector, r\_eigenvalue, r\_eigenvector, c)

**Результаты и их анализ**

A number of numbers on a white background

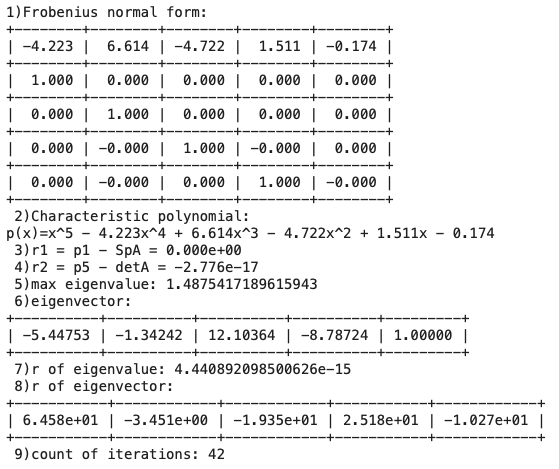
Description automatically generated

Матрица А:

A number and line with numbers

Description automatically generated with medium confidence

Матрица А.T @ A:



Результаты:

A line of numbers and symbols

Description automatically generated

Метод Данилевского является точным методом, это подтверждает маленькая погрешность, близкая к машинной ошибке.