Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Методы численного анализа

**ОТЧЁТ**

к лабораторной работе

на тему

Нахождение собственных значений и собственных векторов

Выполнил: студент группы 253502

Шишко Виктор Викторович

Проверил: Анисимов Владимир Яковлевич

Минск 2023

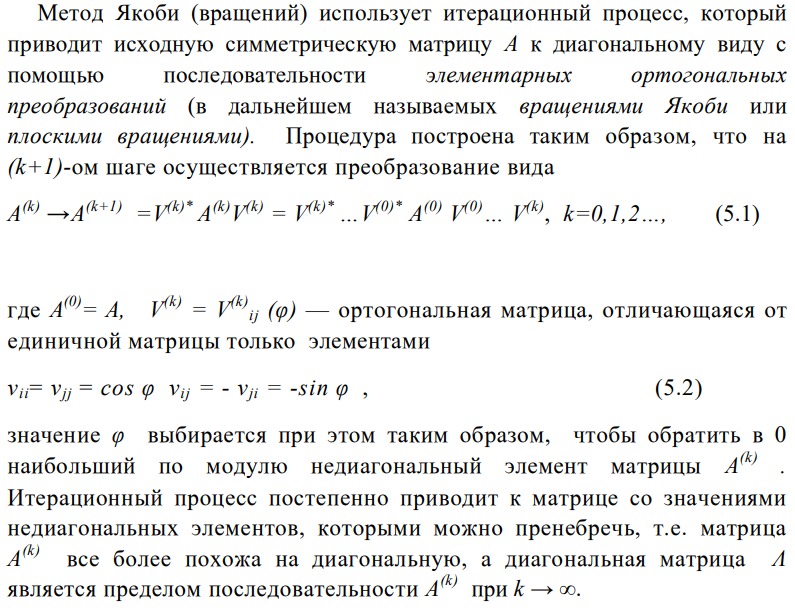
**Цели выполнения задания**

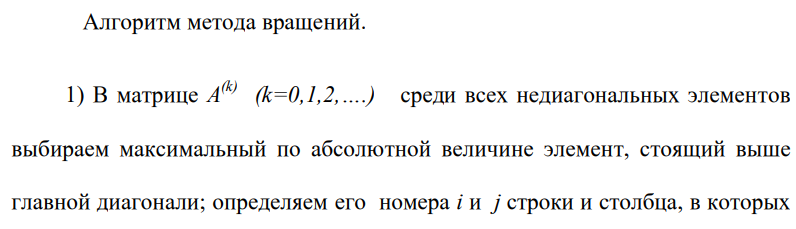
1) Рассмотреть методы вычисления собственных значений и векторов

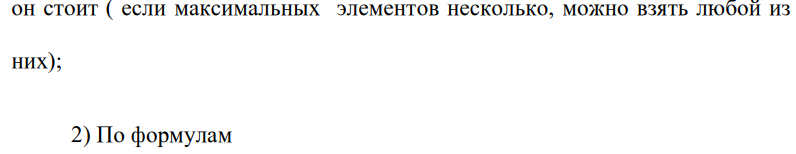
2) Реализовать методы вращений Якоби

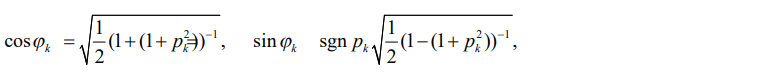
3) Реализовать тестовые примеры

**Краткие теоретические сведения\**



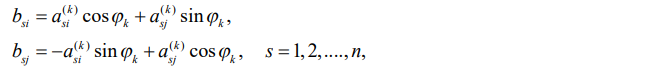


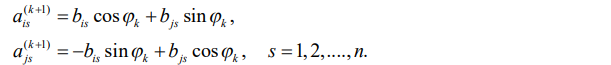




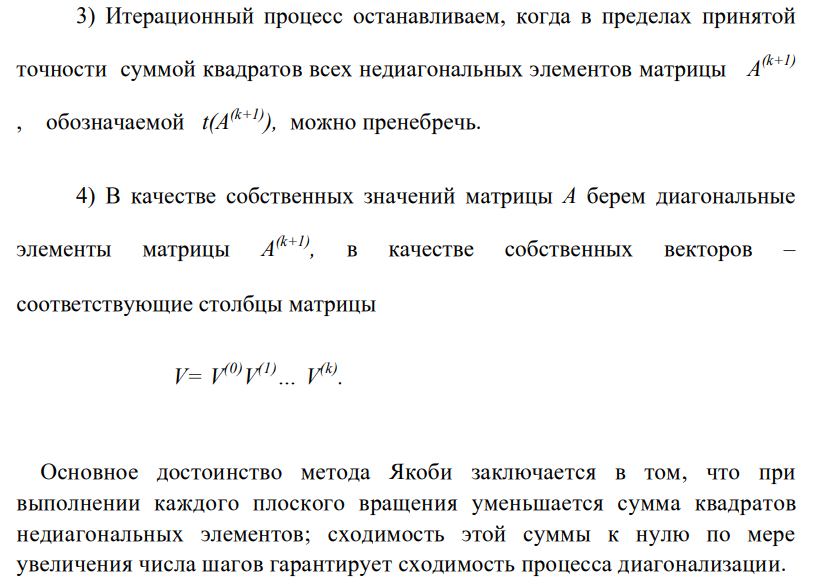










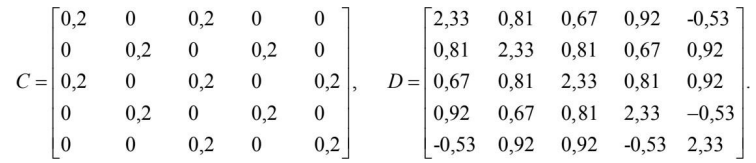


**ПРОГРАММНАЯ РЕАЛИЗАЦИЯ**

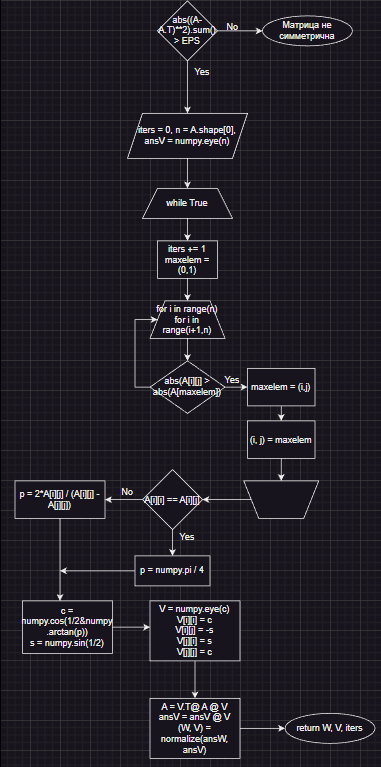
Вариант 14

С точностью до 0,0001 вычислить собственные значения и собственные векторы матрицы A,

где A = kC + D, A – исходная матрица для расчёта, k – номер варианта, матрицы C, D заданы ниже



1. Jacobi



def normalize(W, V):

    V = numpy.array([(-i if i[0] < 0 else i) for i in V.T]).T

    (W, V) = list(zip(\*(sorted(list(zip(W, V.T)), key=lambda t: t[0]))))

    W = numpy.array(W)

    V = numpy.array(V).T

    return (W, V)

def Jacobi(A):

    if abs((A - A.T) \*\* 2).sum() > EPS:

        raise ValueError("Матрица А не симметрична")

    iters = 0

    n = A.shape[0]

    ansV = numpy.eye(n)  # единичная матрица

    while True:

        iters += 1

        # поиск индексов максимального элемента

        maxelem = (0, 1)

        for i in range(n):

            for j in range(i + 1, n):

                if abs(A[i][j]) > abs(A[maxelem]):

                    maxelem = (i, j)

        (i, j) = maxelem

        # угол

        if A[i][i] == A[j][j]:

            p = numpy.pi / 4

        else:

            p = 2 \* A[i][j] / (A[i][i] - A[j][j])

        # вычисляем значения косинусов и синусов для матрицы поворота V

        c = numpy.cos(1 / 2 \* numpy.arctan(p))

        s = numpy.sin(1 / 2 \* numpy.arctan(p))

        # инициализируем матрицу поворта V

        V = numpy.eye(n)

        V[i][i] = c

        V[i][j] = -s

        V[j][i] = s

        V[j][j] = c

        # обновляем матрицу А

        A = V.T @ A @ V

        ansV = ansV @ V

        if abs(A - numpy.diag(numpy.diag(A))).sum() < EPS:

            ansW = numpy.diag(A)  # собственные значения - диагональные эл-ты А

            break

(W, V) = normalize(ansW, ansV)

    # W - собственные значения

    # V - собственные вектора

    return W, V, iters

def Step(A):

    if abs((A - A.T) \*\* 2).sum() > EPS:

        raise ValueError("Матрица А не симметрична")

    r = numpy.ones(len(A))

    iters = 0

    while True:

        iters += 1

        oldu = (r.T @ A @ r) / (r.T @ r)

        r = (A @ r) / numpy.sqrt(sum((A @ r) \*\* 2))

        u = (r.T @ A @ r) / (r.T @ r)

        if abs(u - oldu) < EPS:

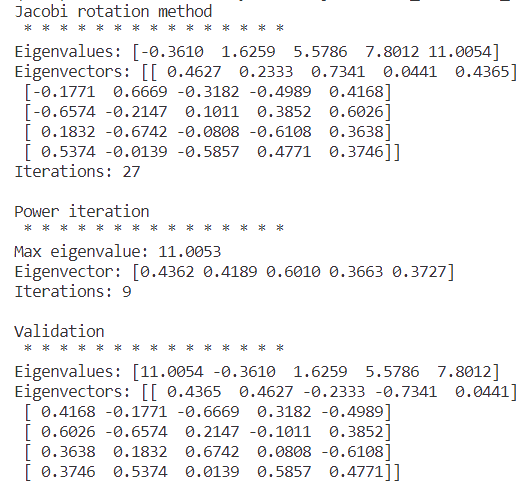
            break

    # u - максимальное собственное значение

    # v - собственный вектор соответствующий макс. значению

    return u, r, iters

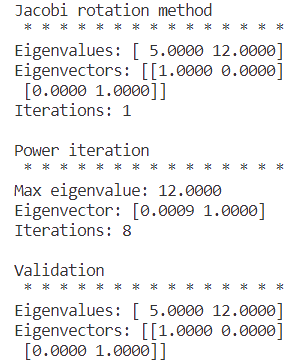
**Результаты работы программной реализации**



**Реализация тестовых примеров**

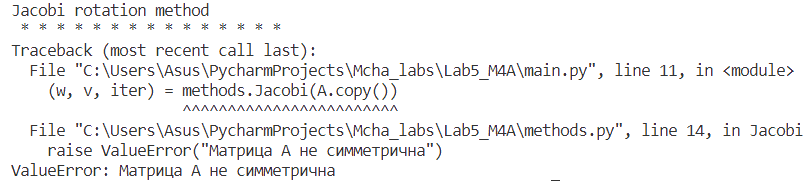
1. С точностью 0.0001 вычислить собственные значения и собственные 24 вектора матрицы A

A\_1 = numpy.array([[5, 0], [0, 12]])



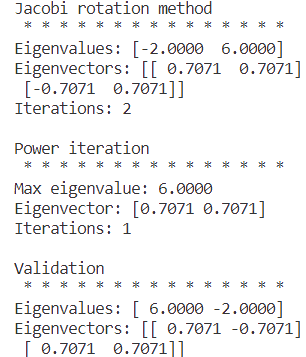
1. С точностью 0.0001 вычислить собственные значения и собственные 24 вектора матрицы A

A\_2 = numpy.array([[2, 7], [8, 10]])(несимметричная матрица)



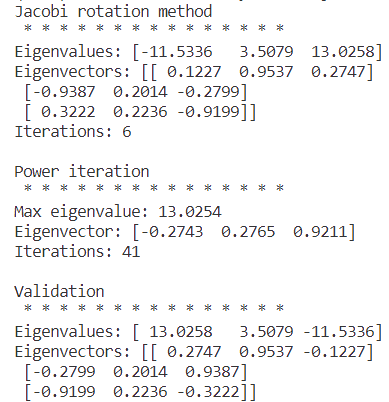
1. С точностью 0.0001 вычислить собственные значения и собственные 24 вектора матрицы A

A\_3 = numpy.array([[2, 4], [4, 2]])



1. С точностью 0.0001 вычислить собственные значения и собственные 24 вектора матрицы A

A\_4 = numpy.array([[4, 1, -3], [1, -9, 7], [-3, 7, 10]])



**Выводы**

В ходе проделанной работы, были рассмотрены матрицы вращения в n мерных пространствах, а также их применение в различных алгоритмах. Одним из них является метод вращения Якоби, который позволяет вычислять все собственные значения для симметричных матриц. Также проведена сверка с numpy методом для определения собственных векторов и значений