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

Вычисление собственных значений и векторов

Выполнил: студент группы 153503

Кончик Денис Сергеевич

Проверил: Анисимов Владимир Яковлевич

Минск 2022

**Содержание**

[1. ЦЕЛЬ РАБОТЫ 3](#_Toc117623160)

[2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ 4](#_Toc117623161)

[3. АЛГОРИТМ РЕШЕНИЯ 7](#_Toc117623162)

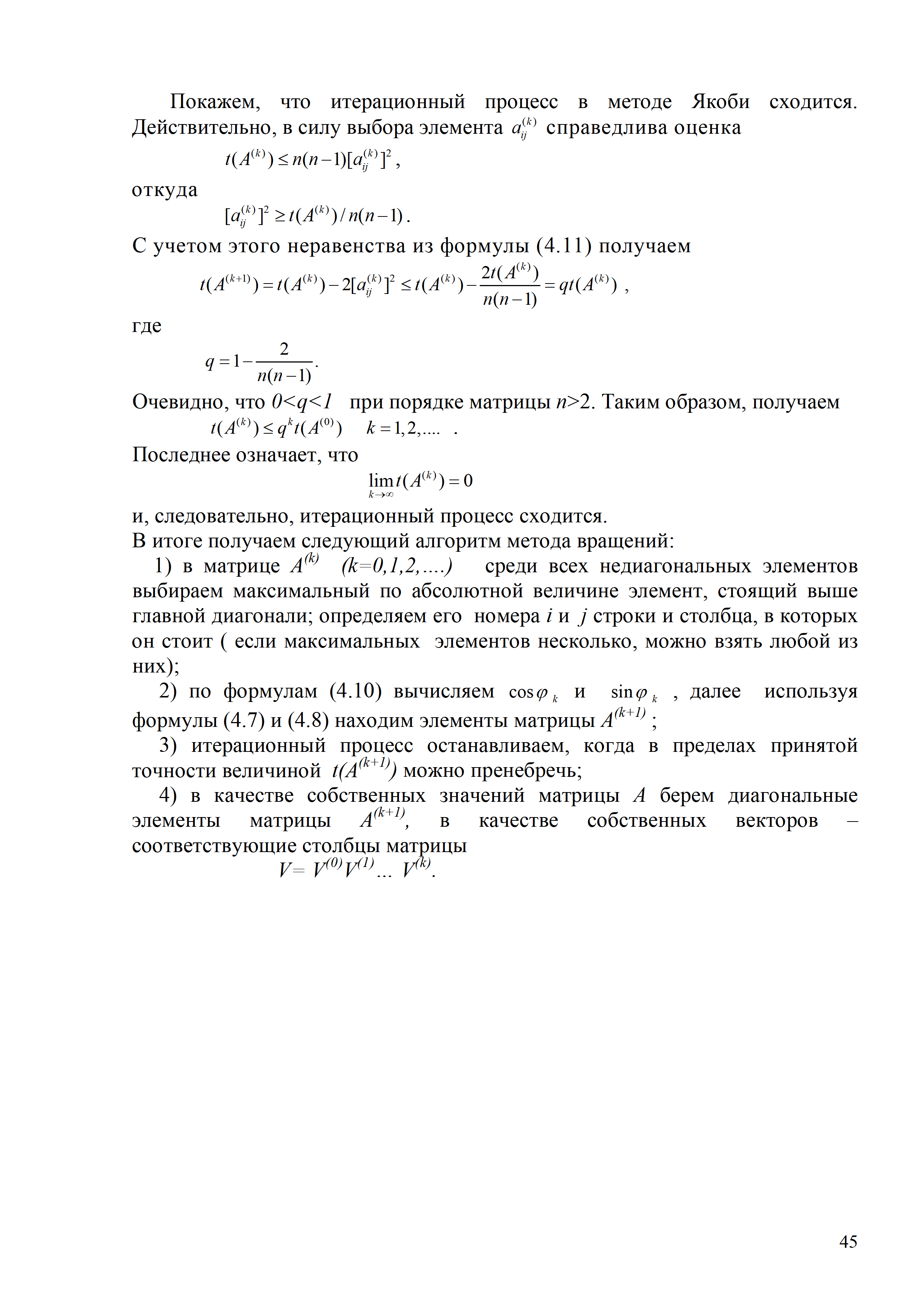
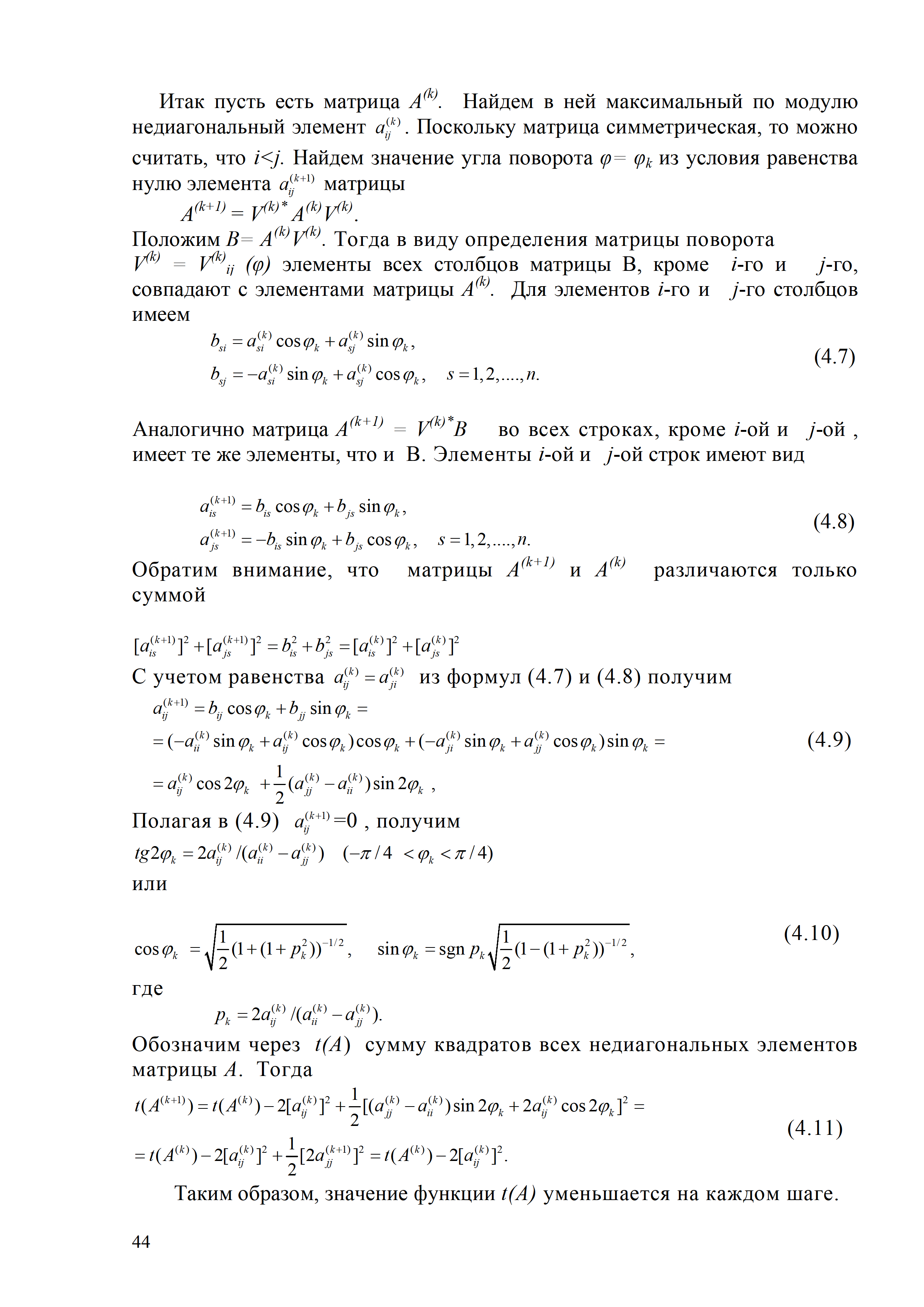
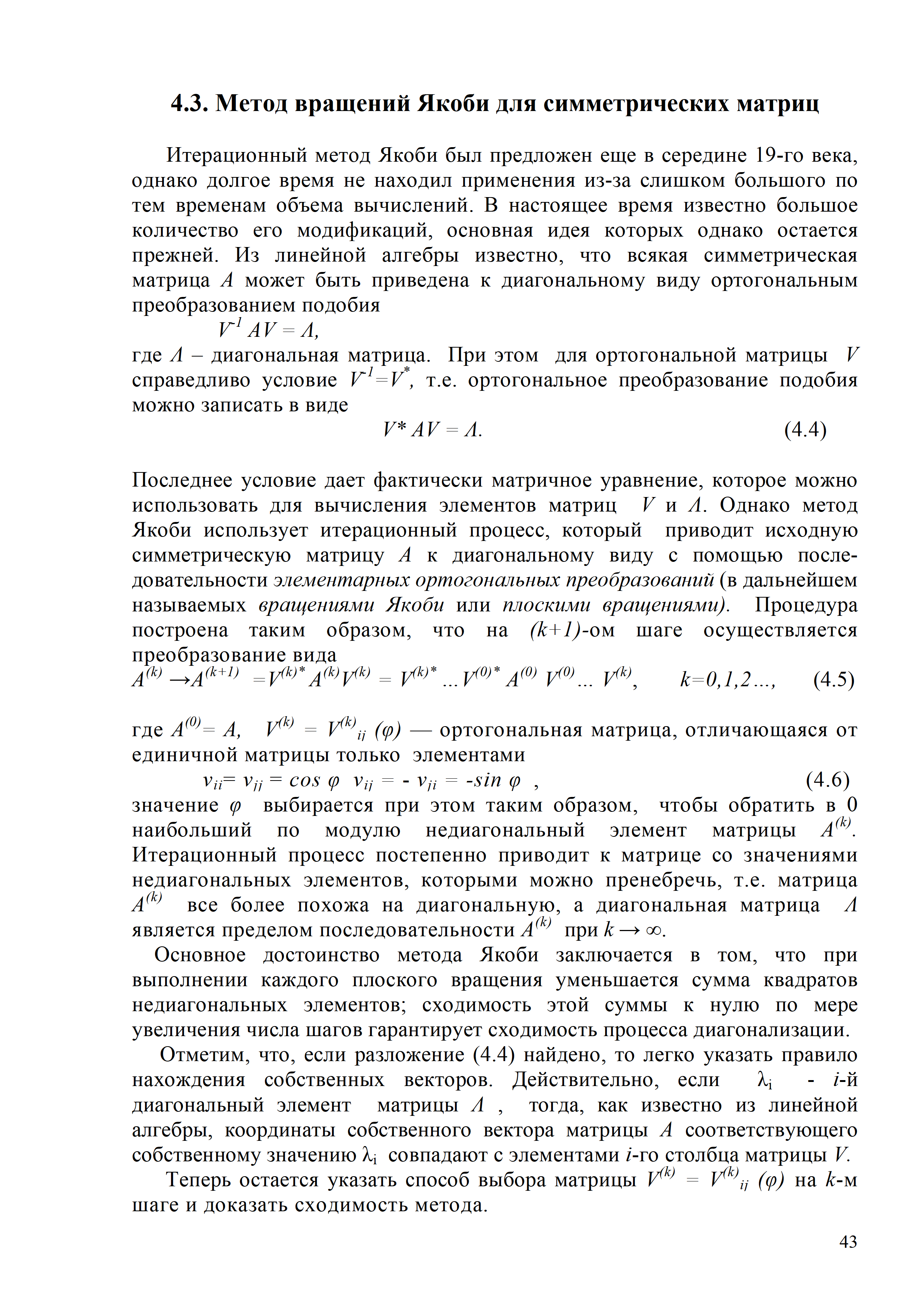
[4. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ 8](#_Toc117623163)

[5. ТЕСТОВЫЕ ПРИМЕРЫ 10](#_Toc117623164)

[6. ЗАДАНИЕ 14](#_Toc117623165)

[7. ВЫВОД 15](#_Toc117623166)

1. **ЦЕЛЬ РАБОТЫ**
2. Изучить метод вращений Якоби для симметрических матриц.
3. Составить алгоритм и программу нахождения собственных значений и собственных векторов методом Якоби.
4. Проверить правильность работы программы на тестовых примерах.
5. Решить задание заданного варианта.
6. **ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**



# **АЛГОРИТМ РЕШЕНИЯ**



# **ПРОГРАММНАЯ РЕАЛИЗАЦИЯ**

import numpy

eps = 0.0001

def CreateMatrix():

k = 10

C = numpy.array([

[0.2, 0.0, 0.2, 0.0, 0.0],

[0.0, 0.2, 0.0, 0.2, 0.0],

[0.2, 0.0, 0.2, 0.0, 0.2],

[0.0, 0.2, 0.0, 0.2, 0.0],

[0.0, 0.0, 0.2, 0.0, 0.2]

])

D = numpy.array([

[2.33, 0.81, 0.67, 0.92, -0.53],

[0.81, 2.33, 0.81, 0.67, 0.92],

[0.67, 0.81, 2.33, 0.81, 0.92],

[0.92, 0.67, 0.81, 2.33, -0.53],

[-0.53, 0.92, 0.92, -0.53, 2.33]

])

return k \* C + D

def GetSumOfSqsFromNodiag(matrix):

sum = 0.0

for i in range(len(matrix)):

for j in range(len(matrix)):

if i != j:

sum += matrix[i][j] \*\* 2

return sum

def GetPosOfMaxFromNodiag(matrix):

max = 0.0

row = 0

column = 0

for i in range(len(matrix)):

for j in range(i + 1, len(matrix)):

if abs(matrix[i][j]) > max:

max = abs(matrix[i][j])

row = i

column = j

return row, column

def GetPhi(matrix, point):

if matrix[point[0]][point[0]] == matrix[point[1]][point[1]]:

return numpy.pi / 4

else:

p = 2 \* matrix[point[0]][point[1]] / (matrix[point[0]][point[0]] - matrix[point[1]][point[1]])

res = 0.5 \* numpy.arctan(p)

return res

def GetR(size, pos, phi):

R = numpy.eye(size)

R[pos[0]][pos[0]] = R[pos[1]][pos[1]] = numpy.cos(phi)

R[pos[0]][pos[1]] = -numpy.sin(phi)

R[pos[1]][pos[0]] = numpy.sin(phi)

return R

def Jacobi(matrix):

V = numpy.eye(len(matrix))

while GetSumOfSqsFromNodiag(matrix) >= eps:

posOfMax = GetPosOfMaxFromNodiag(matrix)

phi = GetPhi(matrix, posOfMax)

R = GetR(len(matrix), posOfMax, phi)

V = V @ R

matrix = R.T @ matrix @ R

#print(matrix)

eigenValues = []

for i in range(len(matrix)):

eigenValues.append(matrix[i][i])

return numpy.array(eigenValues), V.T

def main():

numpy.set\_printoptions(precision = 4, suppress = True, floatmode = "fixed")

A = CreateMatrix()

eigenValues, eigenVectors = Jacobi(A)

print("Исходная матрица: \n", A)

print("\nСобственные значения: \n", eigenValues)

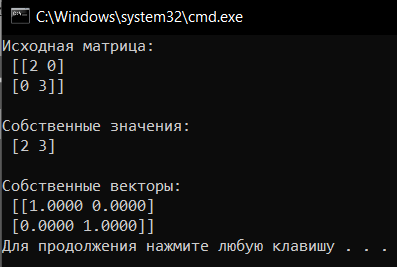
print("\nСобственные векторы: \n", eigenVectors)

main()

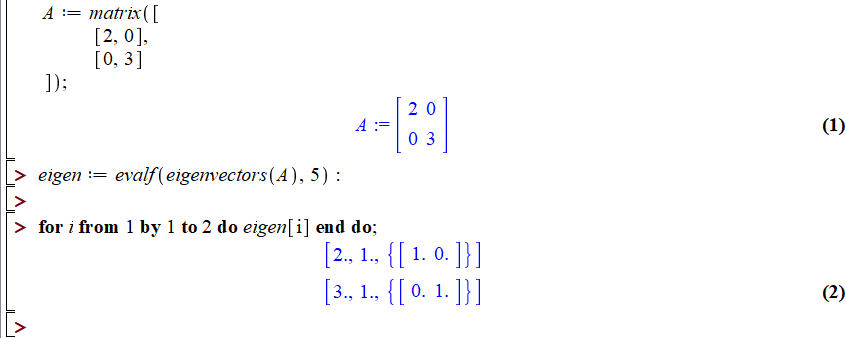
1. **ТЕСТОВЫЕ ПРИМЕРЫ**

Тестовый пример 1

Вычислить с точностью 0.0001 собственные значения и собственные векторы матрицы.

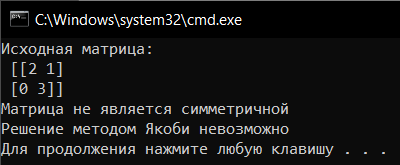
Результат работы программы:

Проверка в Maple:



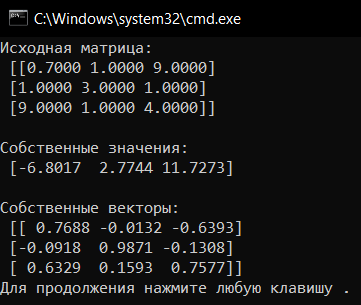
Тестовый пример 2

Вычислить с точностью 0.0001 собственные значения и собственные векторы матрицы.

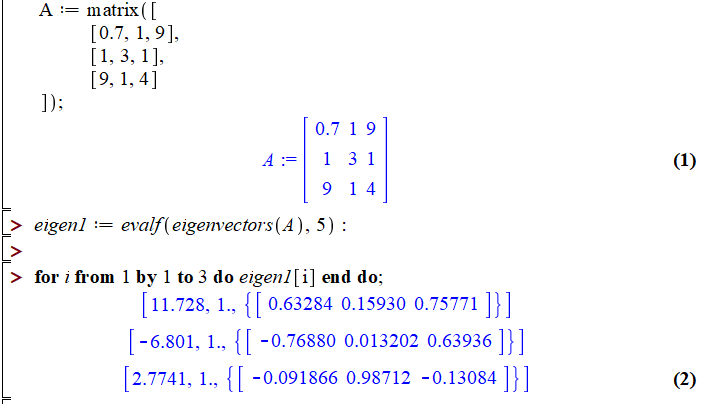
Результат работы программы:

Тестовый пример 3

Вычислить с точностью 0.0001 собственные значения и собственные векторы матрицы.

Результат работы программы:

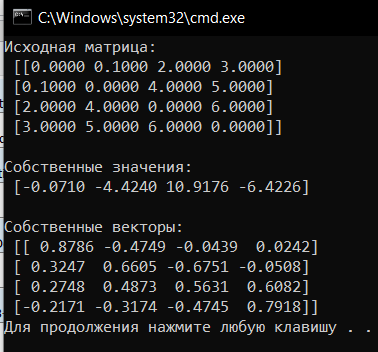
Проверка в Maple:

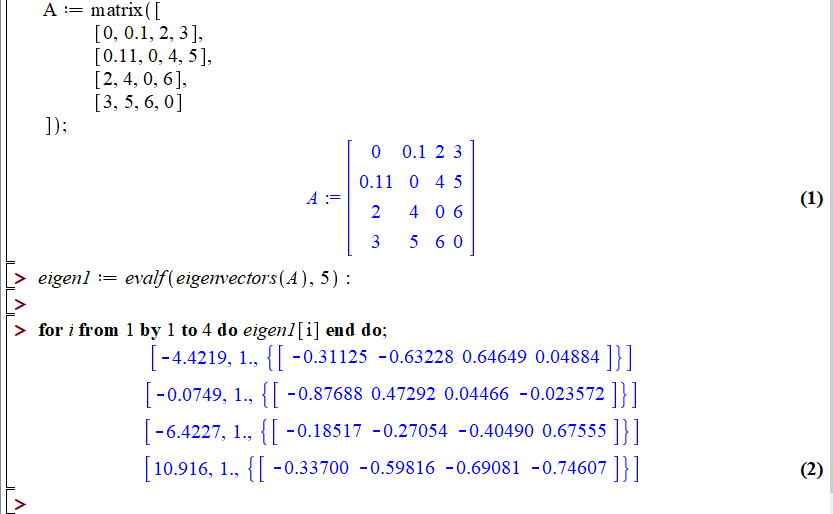


Тестовый пример 4

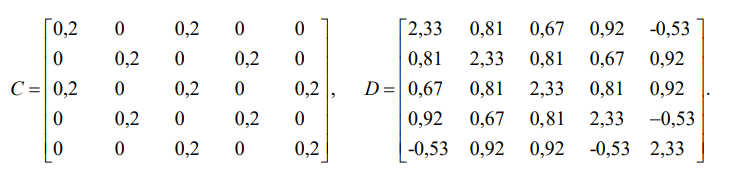
Вычислить с точностью 0.0001 собственные значения и собственные векторы матрицы.

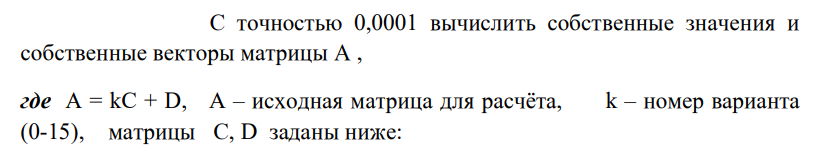
3 5 6 0

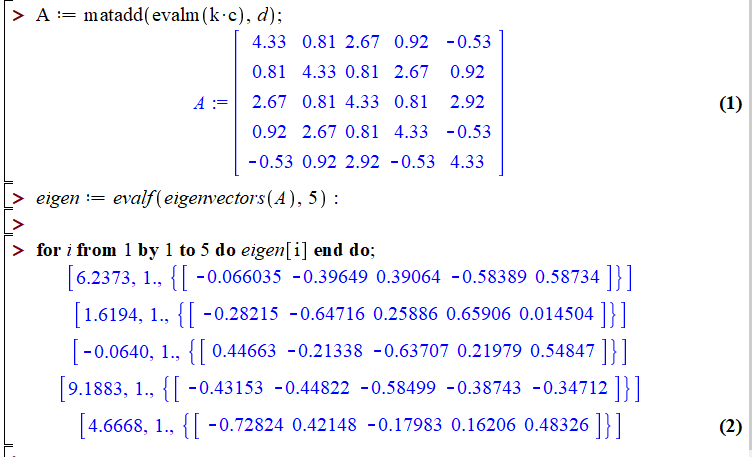
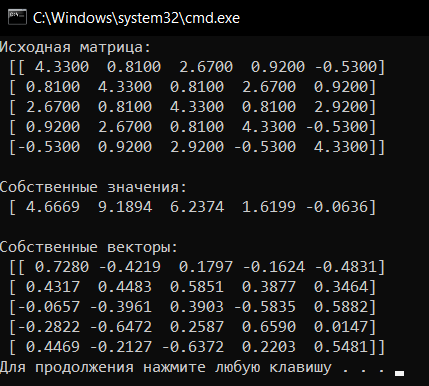
Результат работы программы:

Проверка в Maple:

1. ЗАДАНИЕ

Вариант 10, k = 10





# **ВЫВОД**

Таким образом, в ходе выполнения лабораторной работы был изучен методы вращений Якоби для симметричных матриц, составлен алгоритм и программа нахождения собственных значений и собственных векторов методом Якоби, проверена правильность работы программы на тестовых примерах, численно решено задание заданного варианта.