**Лабораторная работа №10**

«Метод вращений»

выполнил Пажитных Иван, 2-й курс, 1-я группа

**1) 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.

**2)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 .

На каждом шаге итерации строиться матрица

,

где , можно находить по формулам:

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

**3) Листинг программы**

eps = 10 \*\* (-15)

a = **array**(A)

At = a.**transpose**() *# находим Аt*

a = **dot**(At, a) *# перемножаем А на Аt, теперь А – симметрическая*

E, U = identity(n), identity(n)

k = 0

ak = a *# копия А*

**while** (True): *# итерационный процесс*

L = **tril**(ak)

temp = **absolute**(ak - L) *# получаем верхний треугольник, без диагонали*

sigma = **sum**([**abs**(el) \*\* 2 **for** el **in** temp]) *# считаем сумму квадратов недиагональных*

**if** (sigma <= eps): *# условие итерирования*

**break**

i, j = **unravel\_index**(temp.**argmax**(), temp.**shape**) *# индексы максимального*

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

uk = **identity**(n)

uk[i][i], uk[i][j], uk[j][j], uk[j][i] = **cos**(alpha), -**sin**(alpha), **cos**(alpha), **sin**(alpha)

ak = **dot**(**dot**(uk.**transpose**(), ak), uk) *# Ukt\*A\*Uk*

U = **dot**(U, uk) *# U\*Uk*

k += 1

lmax = **max**(ak.**diagonal**()) *# max lambda*

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

x /= **max**(x) *# нормируем*

r = **dot**(a, x) - lmax \* x *# находим вектор невязки*

rnorm = **norm**(r, 1) *# находим норму невязки*

p = [4.58801522, -7.82119475, 6.11344651, -2.15665219, 0.2685558] *# из Данилевского*

p.**insert**(0, -1) *# считаем невязку собственного многочлена*

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

**4) Результат и его анализ**

Симметрическая *АAT*:

[[ 0.70536135 0.01441237 0.13398766 -0.08030921 0.5676231 ]  
 [ 0.01441237 1.22673234 -0.00165256 0.11340719 0.05855825]  
 [ 0.13398766 -0.00165256 0.77976056 -0.21682262 0.2943685 ]  
 [-0.08030921 0.11340719 -0.21682262 0.79926611 -0.0500214 ]  
 [ 0.5676231 0.05855825 0.2943685 -0.0500214 1.07689486]]

Матрица :

[[ 2.74152550e-01 -7.02078740e-16 -1.69795744e-09 9.92616735e-24

-2.51592413e-14]

[ -6.96426697e-16 1.26253884e+00 -3.28971647e-11 -8.73124257e-09

-4.72955349e-11]

[ -1.69795740e-09 -3.28971606e-11 5.50498738e-01 -5.19529740e-14

1.98523347e-23]

[ 2.98254727e-17 -8.73124262e-09 -5.19462112e-14 8.57845170e-01

-2.66909835e-21]

[ -2.50773971e-14 -4.72955215e-11 -5.25380656e-17 8.65306331e-17

1.64297992e+00]]

Коэффициенты собственного многочлена *P(*:

[ 4.58801522 -7.82119475 6.11344651 -2.15665219 0.2685558 ]

Собственные значения - диагональные :

[ 0.27415255 1.26253884 0.55049874 0.85784517 1.64297992]

Максимальное собственное

1.64297992228

Матрица:

[[ 0.75469032 -0.00518699 -0.32156456 0.23944754 0.51930408]

[ 0.00890354 0.95274993 -0.08876095 -0.28144888 0.07138854]

[ 0.21314597 -0.10890943 0.74137613 -0.49970174 0.37863853]

[ 0.14401974 0.28119084 0.5822991 0.72686392 -0.18107078]

[-0.60348186 0.03620937 -0.00263148 0.29226428 0.74099473]]

Собственный вектор матрицы *А —*

[ 0.7008202 0.09634149 0.51098681 -0.24436177 1. ]

Количество итераций *k*:

25

Вектор невязки *r*:

[ 3.06199510e-13 -6.08115225e-11 6.94433400e-12 -1.79525839e-11

-2.28994601e-12]

Норма *||r||*:

8.83045858657e-11

Невязка

1.89213148483e-08

Эпсилон

1e-15

С помощью метода вращений мы нашли максимальное все собственные значения и соответствующий им спектр с точностью порядка 10-1 за 25 итераций для эпсилона порядка 10-15. Невязка собственного многочлена также довльно близка к нулю (порядка 10-8) что означает, что собственное значение также найдено правильно. Собственное значение и собственный вектор также совпадают с получеными ранее методами Крылова и Данилевского. Сравнивая со степенным методом имеем гораздо более высокую скорость сходимости (в СМ 105 итераций), при незначительном ухудшении точности (в СМ невязка порядка 10-13). Построенный итерационный процесс является сходящимся, так как