Министерство образования Республики Беларусь

Учреждение образования “Белорусский государственный университет информатики и радиоэлектроники”

Факультет компьютерных систем и сетей

кафедра Информатики

Дисциплина: Методы численного анализа

ОТЧЕТ

к лабораторной работе

на тему:

“Вычисление собственных значений и векторов”

БГУИР КП 1-40 04 01

Выполнил: студент гр. 953505

Красовский В.Ю.

Проверил: доцент кафедры информатики Анисимов В.Я

Минск 2021

Вариант 9

Цели работы:

Освоить методы вычисления собственных значений и векторов.

Краткие теоретические сведения

Метод Якоби (вращений) использует итерационный процесс, который приводит исходную симметрическую матрицу A к диагональному виду с помощью последовательности элементарных ортогональных преобразовании. Процедура построена таким образом, что на (k+1)-ом шаге осуществляется преобразование вида

$$A^{(k)} \to A^{(k+1)} = V^{(k)*} A^{(k)} V^{(k)} = V^{(k)*} \dots V^{(0)*} A^{(0)} V^{(0)} \dots V^{(k)}, \; k=0,1,2\dots, \qquad (5.1)$$

Где

$A^{(0)} = A, \quad V^{(k)} = V^{(k)}{}_{ij}\,(\varphi)$

ортогональная матрица, отличающаяся от единичной матрицы только элементами

$v_{ii} = v_{jj} = cos\,\varphi \;\; v_{ij} = -\,v_{ji} = -sin\,\varphi$,

значение φ выбирается при этом таким образом, чтобы обратить в 0 наибольший по модулю недиагональный элемент матрицы A(k). Итерационный процесс постепенно приводит к матрице со значениями недиагональных элементов, которыми можно пренебречь, т.е. матрица A(k) все более похожа на диагональную, а диагональная матрица A является пределом последовательности A(k) при k → ꝏ

Алгоритм метода вращений.

1) В матрице A(k) (К=0, 1,2,....) среди всех недиагональных элементов выбираем максимальный по абсолютной величине элемент, стоящий выше главной диагонали, определяем его номера i и j строки и столбца, в которых он стоит (если максимальных элементов несколько, можно взять любой из них);

2) По формулам

$$tg2\varphi_k = 2a_{ij}^{(k)} / (a_{ii}^{(k)} - a_{jj}^{(k)}) \quad (-\pi/4 < \varphi_k < \pi/4)$$

или

$$\cos\varphi_k = \sqrt{\frac{1}{2}(1+(1+p_k^2))^{-1/2}}, \quad \sin\varphi_k = \operatorname{sgn} p_k \sqrt{\frac{1}{2}(1-(1+p_k^2))^{-1/2}},$$

Где

$$p_k = 2a_{ij}^{(k)} / (a_{ii}^{(k)} - a_{jj}^{(k)}).,$$

вычисляем $\cos\varphi_k$ и $\sin\varphi_k$, получаем матрицу $V^{(k)} = V^{(k)}_{ij}(\varphi_k)$.

3) По формулам

$$b_{si} = a_{si}^{(k)}\cos\varphi_k + a_{sj}^{(k)}\sin\varphi_k,$$

$$b_{sj} = -a_{si}^{(k)}\sin\varphi_k + a_{sj}^{(k)}\cos\varphi_k, \quad s = 1,2,....,n,$$

$$a_{is}^{(k+1)} = b_{is}\cos\varphi_k + b_{js}\sin\varphi_k,$$

$$a_{js}^{(k+1)} = -b_{is}\sin\varphi_k + b_{js}\cos\varphi_k, \quad s = 1,2,....,n.$$

Находим элементы матрицы A(k-1)

4) Итерационный процесс останавливаем, когда в пределах принятой точности суммой квадратов всех недиагональных элементов матрицы A(k+1) ,обозначаемой t(A(k+1)), можно пренебречь.

5) В качестве собственных значений матрицы A берем диагональные элементы матрицы A(k+1) качестве собственных векторов — соответствующие столбцы матрицы

$$V = V^{(0)}V^{(1)}\ldots V^{(k)}.$$

Исходные данные:

ЗАДАНИЕ 5. С точностью 0,0001 вычислить собственные значения и собственные векторы матрицы A ,

где A = kC + D, A – исходная матрица для расчёта, k – номер варианта (0-15), матрицы C, D заданы ниже:

$$C=\begin{bmatrix} 0,2 & 0 & 0,2 & 0 & 0 \\ 0 & 0,2 & 0 & 0,2 & 0 \\ 0,2 & 0 & 0,2 & 0 & 0,2 \\ 0 & 0,2 & 0 & 0,2 & 0 \\ 0 & 0 & 0,2 & 0 & 0,2 \end{bmatrix}, \quad D=\begin{bmatrix} 2,33 & 0,81 & 0,67 & 0,92 & -0,53 \\ 0,81 & 2,33 & 0,81 & 0,67 & 0,92 \\ 0,67 & 0,81 & 2,33 & 0,81 & 0,92 \\ 0,92 & 0,67 & 0,81 & 2,33 & -0,53 \\ -0,53 & 0,92 & 0,92 & -0,53 & 2,33 \end{bmatrix}.$$

k = 9 A = k*C +D

Результаты выполнения программы:

```
Matrix a
[[ 4.13  0.81  2.47  0.92 -0.53]
 [ 0.81  4.13  0.81  2.47  0.92]
 [ 2.47  0.81  4.13  0.81  2.72]
 [ 0.92  2.47  0.81  4.13 -0.53]
 [-0.53  0.92  2.72 -0.53  4.13]]
```

```
Собственные значения
[4.4152881366355095, 1.6181281286526432, 0.008087191359354358, 8.738370178559293,
5.8701263647931965]
Собственные векторы
[[ 0.72049373  0.29774967 -0.44124486  0.43059322 -0.11011975]
 [-0.45397369  0.64022669  0.22431536  0.45611541 -0.35448418]
 [ 0.21002286 -0.27206654  0.62974081  0.58051221  0.38510037]
 [-0.19400448 -0.65361987 -0.23099949  0.39367845 -0.5716641 ]
 [-0.43937379 -0.01468977 -0.55231453  0.33848349  0.62218363]]

Проверка средствами numpy
Собственные значения
[8.73837202e+00 8.08604830e-03 1.61812746e+00 4.41528811e+00
 5.87012637e+00]
Собственные векторы
[[ 0.4307813   0.44107769 -0.29761634 -0.72053716 -0.11013019]
 [ 0.45626334 -0.22447268 -0.64008214  0.45393891 -0.35449978]
 [ 0.58018894 -0.62995423  0.27223814 -0.21006192  0.3850959 ]
 [ 0.39357026  0.23085987  0.65374856  0.19396732 -0.57166046]
 [ 0.33872479  0.55219913  0.01478495  0.43933624  0.622179  ]]
```

Тестовый пример 1:

```
Test1
[[2.2 1.  0.5 2. ]
 [1.  1.3 2.  1. ]
 [0.5 2.  0.5 1.6]
 [2.  1.  1.6 2. ]]

Собственные значения
[5.652028916701109, 1.545418317768501, -1.4200865758311747, 0.22263934136156424]
Собственные векторы
[[ 0.53132575 -0.62894356  0.22198403 -0.52234666]
 [ 0.44582986  0.57261649 -0.51587564 -0.4552125 ]
 [ 0.40893405  0.4855982   0.75731466  0.1531073 ]
 [ 0.59304438 -0.20182849 -0.33327155  0.70462309]]

Проверка средствами numpy
Собственные значения
[ 5.65203233  1.54541834  0.22263593 -1.42008659]
Собственные векторы
[[ 0.53173607  0.62892976  0.52192057  0.22204284]
 [ 0.44619412 -0.57257423  0.45486932 -0.51591032]
 [ 0.40881553 -0.4856538  -0.15344702  0.75727423]
 [ 0.59248411  0.20185762 -0.7050864  -0.33327054]]
```

Тестовый пример 2:

```
Test2
[[ 2.  1. -2.]
 [ 1.  0.  3.]
 [-2.  3. -4.]]

Собственные значения
[2.6118234359851966, 1.602023719357569, -6.213847155342766]
Собственные векторы
[[ 0.96320752  0.05675584  0.26269765]
 [ 0.07132207  0.88842185 -0.45345316]
 [-0.25912245  0.45550563  0.85168666]]

Проверка средствами numpy
Собственные значения
[-6.21384726  2.61182354  1.60202372]
Собственные векторы
[[-0.26259206  0.96323619  0.05675776]
 [ 0.45346098  0.07127059  0.888422  ]
 [-0.85171506 -0.25902999  0.45550512]]
```

Тестовый пример 3:

```
Test3
[[ 1.  2. -1.]
 [ 2.  5. -4.]
 [-1. -4. -7.]]

Собственные значения
[0.20780009401575533, 7.019333155738456, -8.22713324975421]
Собственные векторы
[[ 0.93818186  0.34353253  0.04242881]
 [-0.34102575  0.8963441   0.28331728]
 [ 0.05929789 -0.28027245  0.95808722]]

Проверка средствами numpy
Собственные значения
[ 7.01933753  0.20779572 -8.22713325]
Собственные векторы
[[ 0.34428466  0.93790634  0.04242364]
 [ 0.89607038 -0.34174277  0.28331916]
 [-0.28022481  0.05952787  0.95808689]]
```

Вывод

Была написана программа на языке python для нахождения собственных значений и векторов с помощью метода Якоби(вращений)