Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»
Кафедра информатики

Отчет по лабораторной работе №5
Вычисление собственных значений и векторов

Выполнил:
студент гр. 953501
Кореневский С. А.

Руководитель:
доцент
Анисимов В. Я

Минск 2021

Содержание

Цель выполнения задания: 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 преобразований* (в дальнейшем называемых *вращениями Якоби* или *плоскими вращениями).* 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 $A^{(0)} = A$, $V^{(k)} = V^{(k)}_{ij}(\varphi)$ — ортогональная матрица, отличающаяся от единичной матрицы только элементами

$$v_{ii} = v_{jj} = cos\, \varphi \quad v_{ij} = - v_{ji} = -sin\, \varphi \;, \qquad (5.2)$$

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 Λ является пределом последовательности $A^{(k)}$ при $k \to \infty$.

Алгоритм метода вращений.

1) В матрице $A^{(k)}$ $(k=0,1,2,\dots.)$ 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

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

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 $A^{(k+1)}$, в качестве собственных векторов – соответствующие столбцы матрицы

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

Основное достоинство метода Якоби заключается в том, что при выполнении каждого плоского вращения уменьшается сумма квадратов недиагональных элементов; сходимость этой суммы к нулю по мере увеличения числа шагов гарантирует сходимость процесса диагонализации.

Итеративные алгоритмы решают задачу вычисления собственных значений путём построения последовательностей, сходящихся к собственным значениям. Некоторые алгоритмы дают также последовательности векторов, сходящихся к собственным векторам. Чаще всего последовательности собственных значений выражаются через последовательности подобных матриц, которые сходятся к треугольной или диагональной форме, что позволяет затем просто получить собственные значения. Последовательности собственных векторов выражаются через соответствующие матрицы подобия.

Метод	Применим к матрицам	Результат	Цена за один шаг	Сходимость	Описание
Степенной метод	общего вида	наибольшее собственное значение и соответствующий вектор	$O(n^2)$	Линейная	Многократное умножение матрицы на произвольно выбранный начальный вектор с последующей нормализацией.
Обратный степенной метод	общего вида	ближайшее к μ собственное значение и соответствующий вектор		Линейная	Степенная итерация с матрицей $(A - \mu E)^{-1}$
Метод итераций Рэлея	общего вида	ближайшее к μ собственное значение и соответствующий вектор		Кубическая	Степенная итерация с матрицей $(A - \mu_i E)^{-1}$, где μ_i является отношением Рэлея от предыдущей итерации.
Предобусловленная обратная итерация[6] или LOBPCG[en]	положительно определённая вещественная симметричная	ближайшее к μ собственное значение и соответствующий вектор			Обратная итерация с предобуславливанием (приближённое обращение матрицы A).
Метод деления пополам[7]	вещественная симметричная трёхдиагональная	любое собственное значение		Линейная	Использует метод бисекции для поиска корней характеристического многочлена и свойства последовательности Штурма.
Итерации Лагерра	вещественная симметричная трёхдиагональная	любое собственное значение		Кубическая[8]	Использует метод Лагерра[en] вычисления корней характеристического многочлена и свойства последовательности Штурма.
QR-алгоритм[9]	хессенберга	все собственные значения	$O(n^2)$	Кубическая	Разложение $A = QR$, где Q ортогональная, R — треугольная, затем используется итерация к RQ.
		все собственные значения	$6n^3 + O(n^2)$		
Метод Якоби	вещественная симметричная	все собственные значения	$O(n^3)$	квадратичная	Использует поворот Гивенса в попытке избавиться от недиагональных элементов. Попытка не удаётся, но усиливает диагональ.
Разделяй и властвуй[en]	эрмитова трёхдиагональная	все собственные значения	$O(n^2)$		Матрица разбивается на подматрицы, которые диагонализируются, затем воссоединяются.
		все собственные значения	$(\frac{4}{3})n^3 + O(n^2)$		
Метод гомотопии	вещественная симметричная трёхдиагональная	все собственные значения	$O(n^2)$[10]		Строится вычисляемая гомотопия.
Метод спектральной свёртки[en]	вещественная симметричная	ближайшее к μ собственное значение и соответствующий собственный вектор			Предобусловленная обратная итерация, применённая к $(A - \mu E)^2$
Алгоритм MRRR[11]	вещественная симметричная трёхдиагональная	некоторые или все собственные значения и соответствующие собственные вектора	$O(n^2)$		«Multiple Relatively Robust Representations» — Осуществляется обратная итерация с разложением LDL^T смещённой матрицы.

3. Программная реализация

ЗАДАНИЕ 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:

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

Результат работы программы (k = 6, eps = 0.0001)**:**

```
A:
 [[ 3.53  0.81  1.87  0.92 -0.53]
 [ 0.81  3.53  0.81  1.87  0.92]
 [ 1.87  0.81  3.53  0.81  2.12]
 [ 0.92  1.87  0.81  3.53 -0.53]
 [-0.53  0.92  2.12 -0.53  3.53]]

Вектор собственных значений:  [3.5537 7.3934 4.8776 1.6121 0.2131]
Столбцы - собственные векторы:
 [[ 0.6638  0.4298 -0.2703 -0.3557  0.4184]
 [-0.5433  0.4810 -0.1799 -0.6087 -0.2657]
 [ 0.3221  0.5648  0.3388  0.3205 -0.5998]
 [-0.3070  0.4114 -0.5129  0.6325  0.2709]
 [-0.2572  0.3093  0.7189  0.0141  0.5667]]
Количество итераций:  19

Проверка с помощью встроенной функции:
Вектор собственных значений:  [7.3934 0.2131 1.6121 3.5537 4.8777]
Столбцы - собственные векторы:
 [[-0.4298 -0.4184  0.3552  0.6640 -0.2704]
 [-0.4812  0.2657  0.6089 -0.5429 -0.1798]
 [-0.5648  0.5996 -0.3208  0.3219  0.3391]
 [-0.4113 -0.2708 -0.6324 -0.3074 -0.5129]
 [-0.3091 -0.5670 -0.0140 -0.2573  0.7187]]
```

Результат работы программы (k = 6, eps = 0.01)**:**

```
A:
 [[ 3.53  0.81  1.87  0.92 -0.53]
 [ 0.81  3.53  0.81  1.87  0.92]
 [ 1.87  0.81  3.53  0.81  2.12]
 [ 0.92  1.87  0.81  3.53 -0.53]
 [-0.53  0.92  2.12 -0.53  3.53]]

Вектор собственных значений:  [3.5547 7.3931 4.8770 1.6121 0.2132]
Столбцы - собственные векторы:
 [[ 0.6735  0.4237 -0.2554 -0.3584  0.4161]
 [-0.5348  0.4859 -0.1920 -0.6069 -0.2697]
 [ 0.3196  0.5619  0.3459  0.3245 -0.5977]
 [-0.2918  0.4141 -0.5196  0.6307  0.2751]
 [-0.2704  0.3118  0.7130  0.0104  0.5668]]
Количество итераций:  16

Проверка с помощью встроенной функции:
Вектор собственных значений:  [7.3934 0.2131 1.6121 3.5537 4.8777]
Столбцы - собственные векторы:
 [[-0.4298 -0.4184  0.3552  0.6640 -0.2704]
 [-0.4812  0.2657  0.6089 -0.5429 -0.1798]
 [-0.5648  0.5996 -0.3208  0.3219  0.3391]
 [-0.4113 -0.2708 -0.6324 -0.3074 -0.5129]
 [-0.3091 -0.5670 -0.0140 -0.2573  0.7187]]
```

Результат работы программы (k = 0, eps = 0.0001)**:**

```
A:
 [[ 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]]

Вектор собственных значений:  [4.7525 0.5201 1.6154 1.3146 3.4474]
Столбцы - собственные векторы:
 [[ 0.4476 -0.3507 -0.4001 -0.5835 -0.4196]
 [ 0.5279  0.4204 -0.5831  0.3994  0.2124]
 [ 0.5278  0.4204  0.5839 -0.3995  0.2103]
 [ 0.4476 -0.3507  0.3987  0.5835 -0.4210]
 [ 0.2051 -0.6329  0.0013 -0.0000  0.7466]]
Количество итераций:  20

Проверка с помощью встроенной функции:
Вектор собственных значений:  [4.7525 3.4474 0.5201 1.6154 1.3146]
Столбцы - собственные векторы:
 [[-0.4476 -0.4202 -0.3508 -0.3982 -0.5843]
 [-0.5278  0.2113  0.4204 -0.5843  0.3982]
 [-0.5278  0.2113  0.4204  0.5843 -0.3982]
 [-0.4476 -0.4202 -0.3508  0.3982  0.5843]
 [-0.2051  0.7466 -0.6328  0.0000  0.0000]]
```

4. Выводы

В результате выполнения лабораторной работы, я ознакомился с различными методами вычисления собственных значений и векторов. А также реализовал метод Якоби (вращений), и сравнил количество итераций при разной заданной точности и исходных данных.