

Дисциплина: Численные методы  
Лабораторное задание №3

Отчет

Тема: Численные методы решения спектральных задач  
линейной алгебры.  
Метод обратных итераций.

Выполнили:  
студентки 3 курса 62 группы  
Пахомова П.В., Киселёва О.А.

Проверила:  
старший преподаватель Фролова О.А.

## **1. Постановка задачи**

Метод обратных итераций определения пары с минимальным по модулю собственным значением симметричной матрицы простой структуры.

### **Вариант 4.**

**Указание.** Для решения линейной системы уравнений использовать метод Халецкого решения СЛАУ с ленточными матрицами.

### **Входные параметры основной процедуры:**

$N$  – размерность матрицы;

$A$  – двумерный массив размерности  $N \times N$ ;

$\varepsilon_\lambda$  – точность определения собственного значения;

$\varepsilon_g$  – точность определения собственного вектора;

$M$  – максимально допустимое число итераций.

### **Выходные параметры основной процедуры:**

IER – код завершения;

$\lambda$  – минимальное по модулю собственное значение;

$x$  – собственный вектор, соответствующий собственному значению  $\lambda$ ;

$K$  – число выполненных итераций;

$r$  – мера точности полученной пары  $(\lambda, x)$ .

Требуется написать алгоритм нахождения минимального по модулю собственного значения и собственного вектора, соответствующего минимальному собственному значению.

Если для решения системы уравнений с матрицей  $A$  применяется один из методов LU разложения матрицы  $A$ , то один раз найденное LU разложение используется в дальнейшем.

Необходимо написать тест для средней оценки точности собственных значений, собственных векторов, меры точности, числа итераций.

При записи погрешностей используются 2–3 значащие цифры, не более.

## 2. Теоретическая часть

Зная собственные вектора и собственные значения симметричная матрица может быть получена через перемножение матриц, вам понадобятся следующие формулы и свойства:

Симметричная матрица — это матрица, у которой транспонированная матрица равна исходной:  $A = A^T$ .

Если матрица  $P$  состоит из собственных векторов по столбцам, а матрица  $D$  содержит собственные значения на диагонали и нули вне диагонали, то симметричная матрица  $A$  может быть выражена как  $A = PDP^{(-1)}$ , где:

$P$  — матрица, содержащая собственные векторы по столбцам.

$D$  — матрица, содержащая собственные значения на диагонали и нули вне диагонали.

$P^{(-1)}$  — обратная матрица к матрице  $P$ .

Теперь приведем формулы для объяснения этого процесса:

1. Пусть  $P$  — матрица с собственными векторами по столбцам:

$P = [v_1, v_2, \dots, v_n]$ , где каждый  $v_i$  является собственным вектором.

2. Пусть  $D$  — диагональная матрица с собственными значениями:

$D = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}$ , где  $\lambda_i$  — собственные значения.

3. Тогда произведение  $PDP^{(-1)}$  будет иметь вид:

$PDP^{(-1)} = [v_1, v_2, \dots, v_n] * [[\lambda_1 v_1, \lambda_2 v_2, \dots, \lambda_n v_n]]$

4. Это даст нам симметричную матрицу  $A$ :

$A = PDP^{(-1)}$ ,

которая будет иметь собственные векторы  $v_i$  в качестве столбцов и собственные значения  $\lambda_i$  на диагонали.

Таким образом, через умножение матриц  $P$  (содержащей собственные векторы),  $D$  (содержащей собственные значения на диагонали), и  $P^{-1}$  (обратной матрицы  $P$ ) можно получить симметричную матрицу  $A$ .

Ниже приведено несколько методов определения собственных значений  $\lambda$  и собственных векторов  $x$  для симметричных матриц  $A \in R_{n \times n}$ . Рассмотрение только симметричных матриц объясняется двумя причинами. Во-первых, такие матрицы очень просто устроены с точки зрения спектральной задачи (все собственные значения действительны), во-вторых, симметричные матрицы часто возникают в инженерных расчетах. Кроме того, оставим в стороне случаи кратных собственных значений или совпадения их модулей. Будем считать, что собственные значения пронумерованы в порядке возрастания их модулей.

### Степенной метод (метод прямых итераций)

Степенной метод приспособлен для нахождения наибольшего по модулю собственного значения  $\lambda_n$  и соответствующего ему собственного вектора  $x_n$ . Пусть  $x^{(0)}$  – произвольный вектор из  $R_n$ . Вычисления итерационного процесса ведутся по схеме

$$\begin{cases} v^{(k)} = x^{(k)} / \|x^{(k)}\| \\ x^{(k+1)} = A v^{(k)}, \end{cases} \quad k = 0, 1, 2, \dots \quad (2.1.2)$$

с попутным вычислением чисел

$$\sigma^{(k)} = v^{(k)T} x^{(k+1)}. \quad (2.1.3)$$

Показано что

$$\begin{cases} \sigma^{(k)} \rightarrow \lambda_n \\ v^{(k)} \rightarrow \pm x_n \end{cases}, \quad k \rightarrow \infty. \quad (2.1.4)$$

Замечание 1. Во всех приведенных методах берутся евклидовы нормы

Замечание 2. Если  $x_1, x_2, \dots, x_n$  – ортонормированный базис, составленный из собственных векторов матрицы  $A$ , то вектор начального приближения  $x^{(0)}$  разложим по этому базису:

$$x^{(0)} = \sum_{i=1}^n \xi_i x_i.$$

В степенном методе предполагается, что  $\xi_n = -x_n^T x^{(0)} \neq 0$ , т. е. что  $x^{(0)}$  не ортогонален  $x_n$ .

Замечание 3. Скорость сходимости (2.1.2) зависит от отношения  $\left| \frac{\lambda_{n-1}}{\lambda_n} \right|$ :

$$\begin{aligned} \sigma^{(k)} &= \lambda_n \left[ 1 + O \left( \left| \frac{\lambda_{n-1}}{\lambda_n} \right|^{2k} \right) \right] \\ \nu^{(k)} &= \left( \frac{\lambda_n}{|\lambda_n|} \right)^k \frac{\xi_n}{|\xi_n|} \left[ x_n + O \left( \left| \frac{\lambda_{n-1}}{\lambda_n} \right|^k \right) \right] \end{aligned} \quad (2.1.5)$$

Если  $\left| \frac{\lambda_{n-1}}{\lambda_n} \right|$  отношение близко к единице, то сходимость медленная.  
Последовательность  $\sigma^{(k)}$

всегда сходится быстрее, чем последовательность векторов  $\nu^{(k)}$ .

## Метод обратных итераций

Если матрица  $A$  невырожденная, то наибольшее по модулю собственное значение матрицы  $A^{(-1)}$  будет равно  $1/\lambda_1$ . Итерационная схема (2.1.2), примененная к матрице  $A^{(-1)}$ , имеет вид

$$\begin{cases} \mathbf{v}^{(k)} = \mathbf{x}^{(k)} / \|\mathbf{x}^{(k)}\| \\ \mathbf{x}^{(k+1)} = A^{-1}\mathbf{v}^{(k)} \Leftrightarrow A\mathbf{x}^{(k+1)} = \mathbf{v}^{(k)}, \end{cases} \quad k = 0, 1, 2, \dots \quad (2.3.1)$$

причем  $\sigma^{(k)} = \mathbf{v}^{(k)T} \mathbf{x}^{(k+1)} \rightarrow 1/\lambda_1$ ,  $\mathbf{v}^{(k)} \rightarrow \pm \mathbf{x}_1$  при  $k \rightarrow \infty$ . На каждом итерационном шаге вектор  $\mathbf{x}^{(k+1)}$  находится как решение системы линейных уравнений  $A\mathbf{x}^{(k+1)} = \mathbf{v}^{(k)}$

**Замечание 1.** Скорость сходимости итерационного процесса (с8) зависит от отношения  $\left| \frac{\lambda_1}{\lambda_2} \right|$ :

$$\begin{aligned} \alpha^{(k)} &= \frac{1}{\lambda_1} \left[ 1 + O \left( \left| \frac{\lambda_1}{\lambda_2} \right|^{2k} \right) \right], \\ \mathbf{v}^{(k)} &= \left( \frac{\lambda_1}{|\lambda_1|} \right)^k \frac{\xi_1}{|\xi_1|} \left[ \mathbf{x}_1 + O \left( \left| \frac{\lambda_1}{\lambda_2} \right|^k \right) \right] \rightarrow \pm \mathbf{x}_1. \end{aligned} \quad (2.3.2)$$

Замечание о построении тестовых матриц для решения спектральных задач. Для того чтобы составить симметричную матрицу размерности  $N$ , имеющую заранее известные собственные значения, можно поступить следующим образом.

Пусть  $\Lambda = (\lambda_i) \text{diag}$  – диагональная матрица размерности  $N \times N$ ,

$\lambda_i$  – собственные значения конструируемой матрицы  $A$ ,

$\omega$  – случайным образом сгенерированный и пронормированный вектор

( $|\omega| = 1$ ) размерности  $N$ . Образуем с помощью вектора (столбца)  $\omega$  матрицу Хаусхолдера:

$$H = E - 2\omega\omega^T,$$

являющуюся симметричной и ортогональной. Тогда в качестве тестируемой матрицы можно взять матрицу

$$A = H\Lambda H^T,$$

у которой все собственные значения (элементы диагонали матрицы  $\Lambda$ ) и все соответствующие им собственные векторы (столбцы матрицы  $H$ ) известны.

**Замечание** о выходе из итерационного процесса в степенном методе (варианты 1–9).

Итерационный процесс прекращается, если:

- достигнуты требуемые точности определения собственного значения и собственного вектора;
- число итераций превысило максимально допустимое значение.

Предполагается, что требуемая точность  $\varepsilon_\lambda$  для собственного значения достигнута, если модуль разности двух последовательных приближений стал меньше  $\varepsilon_\lambda$ . Аналогично считается, что точность  $\varepsilon_\xi$  для собственного вектора получена, если абсолютное значение угла между двумя векторами, являющимися последовательными приближениями собственного вектора, меньше  $\varepsilon_\xi$ . Если брать одинаковые значения  $\varepsilon_\lambda$ ,  $\varepsilon_\xi$ , то точность для собственных значений достигается, вообще говоря, быстрее, чем для собственных векторов. Поэтому, несмотря на то, что собственное значение получено с заданной точностью, итерационный процесс продолжается до достижения заданной точности собственного вектора.

**Замечание** о выходном параметре  $r$  – мере точности решения спектральной задачи (варианты 1–9).

Под мерой точности понимается максимальное по модулю отклонение компоненты вектора  $Ax - \lambda x$  от нуля, т. е. первая норма вектора  $Ax - \lambda x$ .

**Замечание** о средней оценке точности, о среднем числе итераций.

Как и в предыдущих заданиях, средняя оценка точности, среднее число итераций – это среднее арифметическое соответствующих значений ряда подобных испытаний. Каждый ряд испытаний состоит не менее чем из 10 испытаний.

Напомним, что в записи оценки точности используются 2–3 значащие цифры, не более.

**Замечание** об обязательных вычислительных экспериментах в вариантах 1–9. Результаты тестирования представляются в виде таблицы.

№ теста	Размерность системы $N$	Диапазон значений $\lambda$	Точность ( $\varepsilon_\lambda = \varepsilon_\xi$ )	Ср. оценка точности собств. значений	Ср. оценка точности собств. векторов	Средняя мера точности $r$	Среднее число итераций
1	...	...	...	...	...	...	...



Тестирование проводится для симметричных матриц простой структуры размерности 10, 30, 50, имеющих собственные значения в диапазонах  $-2 \div 2$ ,  $-50 \div 50$  с разностью  $10^{-5}$ ,  $10^{-8}$ . Минимальное количество строк таблицы равно 12.

### 3. Алгоритм

#### ШАГ 1

Генерируем симметричную матрицу.

Функция `GeneratorSymmetricMatrixWithEigenVectorsAndValues` генерирует симметричную матрицу размера `size` с заданным диапазоном собственные вектора и значения.

Минимальное и максимальное значение для собственных значений передаются в `lambdaMin` и `lambdaMax` соответственно. Если они не заданы, то используются значения `min` и `max`.

Алгоритм:

1. Инициализировать переменные `lambdaMin` и `lambdaMax`.
2. Создать генератор случайных чисел.
3. Заполнить массив `_eigenValuesData` случайными значениями в пределах от `lambdaMin` до `lambdaMax`.
4. Заполнить массив `_eigenVectorsData` случайными значениями в пределах от `min` до `max`.
5. Вычислить симметричную матрицу `_symmetricMatrix` как произведение матриц `_eigenVectorsData` и `_eigenValuesData`.
6. Вычислить обратную матрицу `_inverseEigenVectorsData` для `_eigenVectorsData`.
7. Вычислить `_symmetricMatrix` как произведение `_symmetricMatrix`, `_inverseEigenVectorsData` и `size`.

Функция multiply вычисляет произведение двух матриц matrix1 и matrix2 размера size.

Алгоритм:

1. Создать матрицу result размера size x size, заполненную нулями.
2. Для каждого  $i$  от 0 до size-1 выполнить следующее:
  - а. Для каждого  $j$  от 0 до size-1 выполнить следующее:
    - і. Для каждого  $k$  от 0 до size-1 выполнить следующее:
      1. Добавить в result[i][j] произведение matrix1[i][k] и matrix2[k][j].
3. Вернуть матрицу result.

Функция inverseMatrix вычисляет обратную матрицу  $A^{-1}$  для заданной матрицы A методом Гаусса-Жордана.

Алгоритм:

1. Получить размер матрицы A как n.
2. Создать расширенную матрицу augmentedMatrix размера  $n \times 2n$ , заполненную нулями.
3. Заполнить первые n столбцов augmentedMatrix значениями из матрицы A.
4. Заполнить последние n столбцов augmentedMatrix единицами на диагонали.
5. Для каждого  $i$  от 0 до  $n-1$  выполнить следующее:
  - а. Получить pivot как элемент augmentedMatrix[i][i].
  - б. Для каждого  $j$  от 0 до  $2n-1$  выполнить следующее:
    - і. Разделить augmentedMatrix[i][j] на pivot.
  - с. Для каждого  $j$  от 0 до  $n-1$  выполнить следующее:
    - і. Если  $j$  не равно  $i$ , то получить factor как элемент augmentedMatrix[j][i].
    - іі. Для каждого  $k$  от 0 до  $2n-1$  выполнить следующее:
      1. Вычесть из augmentedMatrix[j][k] произведение factor на augmentedMatrix[i][k].
6. Создать матрицу inverse размера  $n \times n$ , заполненную нулями.
7. Заполнить матрицу inverse элементами из последних n столбцов augmentedMatrix.
8. Вернуть матрицу inverse.

## ШАГ 2

Передать матрицу в метод для нахождения минимального по модулю собственного значения и собственного вектора.

*Входные данные:*

- `_size` - размер квадратной симметричной матрицы `_symmetricMatrix`;
- `_symmetricMatrix` - матрица размером `_size` x `_size`, представленная в виде вектора векторов типа `double`, содержащая элементы матрицы;
- `_givenEigenvectorsE` - заданная точность для собственных векторов;
- `_givenEigenValuesE` - заданная точность для собственных значений;
- `_maxIterationsNumber` - максимальное число итераций при поиске собственных значений и векторов.

*Выходные данные:*

- `_firstMinEigenValue` - первое минимальное по модулю собственное значение матрицы `_symmetricMatrix`;
- `_eigenVectorByFirstMinEigenValue` - собственный вектор, соответствующий первому минимальному по модулю собственному значению матрицы `_symmetricMatrix`;
- `_IterationsNumber` - количество выполненных итераций алгоритма поиска собственных значений и векторов;
- `_resultedEigenvectorsE` - фактическая погрешность для собственных векторов;
- `_resultedEigenValuesE` - фактическая погрешность для собственных значений;
- `r` — мера точности полученной пары  
( `_firstMinEigenValue` , `_eigenVectorByFirstMinEigenValue` )

Значения `_firstMinEigenValue`, `_eigenVectorByFirstMinEigenValue`, `_IterationsNumber`, `_resultedEigenvectorsE` и `_resultedEigenValuesE` вычисляются внутри функции `Solve()`. Значение `r` вычисляется после выполнения функции `Solve()`.

Алгоритм:

1. Инициализировать переменные  $x\_rand$ ,  $k$ ,  $q$ ,  $qPrev$ ,  $maxVecE$ ,  $system$ .
2. Заполнить вектор  $x\_rand$  случайными числами в пределах от -10 до 10.
3. Пока не достигнут критерий останова  
(  $\_resultedEigenValuesE > \_givenEigenValuesE$  или  $|maxVecE| > \_givenEigenVectorsE$ ) и  
 $k$  меньше  $\_maxIterationsNumber$ ,  
выполнять следующее:
  - a. Нормализовать вектор  $x\_rand$  с помощью функции `normalizeVector`.  
 $V = x\_rand$
  - b. Решить систему линейных уравнений  $system$  для вектора  $v$ , полученного на предыдущем шаге.
  - c. Если система успешно решена, то обновить  $x\_rand$  как полученное решение.
  - d. Сохранить прошлое значение  $q$  в  $qPrev$ .  
Вычислить произведение между векторами  $v^T$  и  $x\_rand$  и сохранить его в  $q$ .
  - e. Вычислить первое минимальное собственное значение как  $1/q$ .
  - f. Вычислить максимальную по модулю разность между элементами векторов  $v$  и  $\_eigenVectorByFirstMinEigenValue$  и сохранить ее в  $maxVecE$ .
  - g. Обновить  $\_resultedEigenVectorsE$ ,  $\_resultedEigenValuesE$  и  $\_eigenVectorByFirstMinEigenValue$ .
  - h. Увеличить  $k$  на 1.
4. Вычислить вектор  $\_R$  как произведение матрицы  $\_symmetricMatrix$  и  $\_eigenVectorByFirstMinEigenValue$ , вычесть из него произведение  $\_firstMinEigenValue$  и вектора  $\_eigenVectorByFirstMinEigenValue$ .
5. Вычислить максимальное значение вектора  $\_R$  и сохранить его в  $r$ .

Функция `normalizeVector` нормализует заданный вектор путем деления каждого элемента на длину вектора.

Алгоритм:

1. Вычислить сумму квадратов элементов вектора `vector`.
2. Вычислить длину вектора `magnitude` как корень из суммы квадратов элементов.
3. Создать новый вектор `normalizedVector` размера, равного размеру вектора `vector`.
4. Для каждого элемента вектора `vector` выполнить следующее:
  - а. Разделить элемент на длину вектора и добавить результат в `normalizedVector`.
5. Вернуть `normalizedVector`.

## Результаты тестирования

№ теста	Раз- мер	$\lambda$ диапа- зон	E	Ср оценка собственного значения	Ср оценка собственного вектора	Ср. r	Ср. Кол-во Итераций
1	10	-2:2	1.0e-05	5.811e-06	2.466e-06	-3.919e-07	162
2	10	-2:2	1.0e-08	5.890e-09	5.121e-09	-1.505e-09	99
3	10	-50:50	1.0e-05	6.519e-07	7.751e-06	-5.556e-05	31
4	10	-50:50	1.0e-08	2.564e-09	7.328e-09	-1.974e-08	84
5	30	-2:2	1.0e-05	5.893e-06	2.391e-06	-1.076e-07	21
6	30	-2:2	1.0e-08	4.279e-09	5.406e-09	-5.363e-10	81
7	30	-50:50	1.0e-05	3.038e-06	6.852e-06	-1.844e-05	97
8	30	-50:50	1.0e-08	2.329e-09	6.063e-09	-1.000e-08	116
9	50	-2:2	1.0e-05	6.229e-06	4.982e-07	-2.907e-08	28
10	50	-2:2	1.0e-08	6.902e-09	2.375e-09	-1.614e-10	57
11	50	-50:50	1.0e-05	3.275e-06	6.561e-06	-7.576e-06	30
12	50	-50:50	1.0e-08	3.026e-09	6.843e-09	-8.250e-09	46