

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ — ПРОЦЕССОВ УПРАВЛЕНИЯ

А. С. Еремин, О. С. Фирюлина

Практикум на ЭВМ по численным методам

Тема 4. Алгебраическая проблема собственных значений

Методические указания

Санкт-Петербург
2023

Содержание

Свойства собственных чисел и векторов	2
Прямой и обратный степенные методы	5
Степенной метод	5
Обратные итерации	8
QR-алгоритм	11
LU- и QR-алгоритмы	11
Приведение матрицы к форме Хессенберга	12
Сдвиги и понижение размерности	14
Задание для самостоятельного выполнения	16
Литература	17

Принятые обозначения

- Векторы обозначаются строчными полужирными буквами: \mathbf{x}_1 , $\mathbf{y}^{(0)}$, $\boldsymbol{\lambda}$, а их координаты (скаляры) — соответствующими строчными наклонными буквами с индексами: x_{11} , $y_3^{(0)}$, λ_i и т. д.
- По-умолчанию векторы считаются столбцами. Соответственно, транспонированные векторы, напр. \mathbf{x}_1^T , \mathbf{v}^T , — это строки.
- Матрицы обозначаются прописными полужирными буквами: \mathbf{A} , \mathbf{E} и т. д.
- Скалярное произведение векторов \mathbf{x} и \mathbf{y} обозначается (\mathbf{x}, \mathbf{y}) .

Свойства собственных чисел и векторов

Определение 1. *Собственным числом* и (правым) *собственным вектором* квадратной матрицы $\mathbf{A}_{[n \times n]}$ называются число λ и ненулевой вектор \mathbf{x} , для которых выполняется

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}. \quad (1)$$

С геометрической точки зрения \mathbf{x} — такой вектор, что линейный оператор \mathbf{A} лишь растягивает (или сжимает) его без вращения, а λ — соответствующий коэффициент растяжения (или сжатия). Пару $\{\lambda, \mathbf{x}\}$ называют *собственной парой* матрицы \mathbf{A} . Полный набор собственных чисел матрицы называется её *спектром*.

Простейшим теоретическим методом нахождения всех собственных чисел является решение *характеристического уравнения* матрицы \mathbf{A} :

$$\det(\lambda\mathbf{E} - \mathbf{A}) = 0, \quad (2)$$

где \mathbf{E} — единичная матрица. Векторы после этого находятся как частные решения вырожденных систем (1).

Однако сложность решения уравнения (2) и погрешности, возникающие при этом, приводят к необходимости поиска более эффективных методов решения именно задачи поиска собственных пар (всех или одной) или даже только собственных значений.

Рассмотрим свойства собственных пар, которые будут использованы в дальнейшем при построении и модификации изучаемых методов.

Свойство 1. Собственный вектор определяется с точностью до направления, то есть если $\{\lambda, \mathbf{x}\}$ — собственная пара \mathbf{A} , то и $\{\lambda, \alpha\mathbf{x}\}$ является собственной парой \mathbf{A} для любого числа $\alpha \neq 0$:

$$\mathbf{A}(\alpha\mathbf{x}) = \alpha\mathbf{A}\mathbf{x} = \alpha\lambda\mathbf{x} = \lambda(\alpha\mathbf{x}).$$

Свойство 2. Собственные числа «сдвинутой на μ » матрицы $\mathbf{A} + \mu\mathbf{E}$ «сдвинуты на μ » относительно собственных чисел исходной матрицы \mathbf{A} , а их собственные векторы совпадают:

$$(\mathbf{A} + \mu\mathbf{E})\mathbf{x} = \mathbf{A}\mathbf{x} + \mu\mathbf{x} = \lambda\mathbf{x} + \mu\mathbf{x} = (\lambda + \mu)\mathbf{x}.$$

Свойство 3. Собственные числа обратной матрицы \mathbf{A}^{-1} суть обратные величины к собственным числам исходной матрицы \mathbf{A} (конечно, если \mathbf{A} обратима), а их собственные векторы совпадают:

$$\mathbf{Ax} = \lambda \mathbf{x} \iff \mathbf{x} = \mathbf{A}^{-1} \lambda \mathbf{x} \iff \mathbf{A}^{-1} \mathbf{x} = \lambda^{-1} \mathbf{x}.$$

Свойство 4. Собственными числами диагональных и треугольных матриц являются их диагональные элементы.

Определение 2. *Отношением Рэля*¹ для квадратной матрицы называется функционал

$$\varrho(\mathbf{x}) = \frac{(\mathbf{Ax}, \mathbf{x})}{(\mathbf{x}, \mathbf{x})},$$

определённый для любого ненулевого вектора \mathbf{x} .

Свойство 5. Отношение Рэля для собственного вектора \mathbf{x} — это соответствующее собственное число λ :

$$\varrho(\mathbf{x}) = \frac{(\mathbf{Ax}, \mathbf{x})}{(\mathbf{x}, \mathbf{x})} = \frac{(\lambda \mathbf{x}, \mathbf{x})}{(\mathbf{x}, \mathbf{x})} = \lambda.$$

Свойство 6. Минимум евклидовой нормы вектора $\mathbf{r}(\lambda) = \mathbf{Ax} - \lambda \mathbf{x}$ для любого фиксированного ненулевого вектора \mathbf{x} достигается при $\lambda = \varrho(\mathbf{x})$. Говоря другими словами, если \mathbf{x} приближение к собственному вектору (а \mathbf{r} — его невязка), то $\varrho(\mathbf{x})$ — наилучшее приближение к соответствующему собственному числу.

Определение 3. Матрицы \mathbf{A} и \mathbf{B} называются *подобными*, если существует невырожденная матрица \mathbf{C} такая, что $\mathbf{A} = \mathbf{C}^{-1} \mathbf{B} \mathbf{C}$.

Свойство 7. Собственные числа подобных матриц совпадают, а их собственные векторы связаны преобразованием с матрицей подобия:

$$\mathbf{Ax} = \lambda \mathbf{x} \iff \mathbf{C}^{-1} \mathbf{B} \mathbf{C} \mathbf{x} = \lambda \mathbf{x} \iff \mathbf{B}(\mathbf{C} \mathbf{x}) = \lambda(\mathbf{C} \mathbf{x}),$$

¹ *Джон Уильям Стрэтт, третий барон Рэлей* (англ. John Strutt, 3rd Baron Rayleigh) (1842–1919), британский физик и механик, один из основоположников теории колебаний. Получил Нобелевскую премию по физике в 1904 году, как один из открывших газ аргон. В честь него также называются открытое им рассеивание света на объектах, размеры которых меньше его длины волны (Рэлеевское рассеяние), и теоретически предсказанные им поверхностные акустические волны (волны Рэля).

то есть собственная пара $\{\lambda, \mathbf{x}\}$ матрицы \mathbf{A} соответствует собственной паре $\{\lambda, \mathbf{C}\mathbf{x}\}$ матрицы $\mathbf{B} = \mathbf{C}\mathbf{A}\mathbf{C}^{-1}$.

Определение 4. Матрица \mathbf{A} называется *матрицей простой структуры*, если у неё есть ровно n линейно независимых собственных векторов.

Свойство 8. Матрица \mathbf{A} простой структуры подобна диагональной матрице $\mathbf{\Lambda} = \text{diag}(\lambda_i)$, состоящей из её собственных чисел. При этом столбцами матрицы преобразования $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ являются соответствующие собственные векторы:

$$\mathbf{A}\mathbf{x}_i = \lambda_i\mathbf{x}_i \iff \mathbf{A}\mathbf{X} = (\lambda_1\mathbf{x}_1, \dots, \lambda_n\mathbf{x}_n) = \mathbf{X}\mathbf{\Lambda} \iff \mathbf{\Lambda} = \mathbf{X}^{-1}\mathbf{A}\mathbf{X}.$$

Прямой и обратный степенные методы

Степенной метод

Рассмотрим матрицу $\mathbf{A}_{[n \times n]}$ простой структуры. У неё существует n линейно независимых собственных векторов, представимых в пространстве \mathbb{R}_n как

$$\mathbf{x}_1 = \begin{bmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1n} \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} x_{21} \\ x_{22} \\ \vdots \\ x_{2n} \end{bmatrix}, \quad \dots, \quad \mathbf{x}_n = \begin{bmatrix} x_{n1} \\ x_{n2} \\ \vdots \\ x_{nn} \end{bmatrix}.$$

Пусть эти векторы упорядочены по убыванию абсолютной величины соответствующих собственных чисел, причём наибольшее по модулю собственное число лишь одно:

$$|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|.$$

Возьмём произвольный ненулевой вектор \mathbf{y}_0 и представим его в базисе собственных векторов матрицы \mathbf{A} :

$$\mathbf{y}^{(0)} = c_1 \mathbf{x}_1 + c_2 \mathbf{x}_2 + \dots + c_n \mathbf{x}_n, \quad (3)$$

или покомпонентно

$$y_i^{(0)} = c_1 x_{1i} + c_2 x_{2i} + \dots + c_n x_{ni}, \quad i = 1, \dots, n.$$

При этом будем считать, что $c_1 \neq 0$.

Замечание 1. В маловероятном случае, когда $c_1 = 0$ все последующие выкладки теоретически останутся верны для c_2 и \mathbf{x}_2 вместо c_1 и \mathbf{x}_1 . На практике же c_1 из-за неточностей вычислений после нескольких преобразований можно будет считать очень малым, но всё же ненулевым. Конечно, при обнаружении очень плохой сходимости алгоритма, следует выбрать другой начальный вектор, в котором c_1 будет сравнимым с коэффициентами разложения по другим собственным векторам.

Построим итерационный процесс, начинающийся с $\mathbf{y}^{(0)}$, последо-

вательно умножая векторы слева на матрицу \mathbf{A} :

$$\begin{aligned} \mathbf{y}^{(1)} &= \mathbf{A}\mathbf{y}^{(0)} = c_1\mathbf{A}\mathbf{x}_1 + c_2\mathbf{A}\mathbf{x}_2 + \cdots + c_n\mathbf{A}\mathbf{x}_n, \\ &\dots \\ \mathbf{y}^{(k)} &= \mathbf{A}\mathbf{y}^{(k-1)} = c_1\mathbf{A}^k\mathbf{x}_1 + c_2\mathbf{A}^k\mathbf{x}_2 + \cdots + c_n\mathbf{A}^k\mathbf{x}_n, \\ &\dots \end{aligned}$$

что в силу определения собственных пар матрицы \mathbf{A} равносильно

$$\begin{aligned} \mathbf{y}^{(1)} &= \mathbf{A}\mathbf{y}^{(0)} = c_1\lambda_1\mathbf{x}_1 + c_2\lambda_2\mathbf{x}_2 + \cdots + c_n\lambda_n\mathbf{x}_n, \\ &\dots \\ \mathbf{y}^{(k)} &= \mathbf{A}\mathbf{y}^{(k-1)} = c_1\lambda_1^k\mathbf{x}_1 + c_2\lambda_2^k\mathbf{x}_2 + \cdots + c_n\lambda_n^k\mathbf{x}_n, \\ &\dots \end{aligned} \tag{4}$$

Сравнивая покомпонентно векторы $\mathbf{y}^{(k)}$ и $\mathbf{y}^{(k-1)}$ получим соотношения

$$\begin{aligned} \frac{y_i^{(k)}}{y_i^{(k-1)}} &= \frac{c_1\lambda_1^k x_{1i} + c_2\lambda_2^k x_{2i} + \cdots + c_n\lambda_n^k x_{ni}}{c_1\lambda_1^{k-1} x_{1i} + c_2\lambda_2^{k-1} x_{2i} + \cdots + c_n\lambda_n^{k-1} x_{ni}} = \\ &= \frac{c_1\lambda_1^k x_{1i} \left(1 + \frac{c_2}{c_1} \frac{\lambda_2^k}{\lambda_1^k} \frac{x_{2i}}{x_{1i}} + \cdots + \frac{c_n}{c_1} \frac{\lambda_n^k}{\lambda_1^k} \frac{x_{ni}}{x_{1i}} \right)}{c_1\lambda_1^{k-1} x_{1i} \left(1 + \frac{c_2}{c_1} \frac{\lambda_2^{k-1}}{\lambda_1^{k-1}} \frac{x_{2i}}{x_{1i}} + \cdots + \frac{c_n}{c_1} \frac{\lambda_n^{k-1}}{\lambda_1^{k-1}} \frac{x_{ni}}{x_{1i}} \right)}, \end{aligned}$$

верные для всех i , для которых $x_{1i} \neq 0$. В силу того, что $\left| \frac{\lambda_j}{\lambda_1} \right| < 1$, $j = 2, \dots, n$ все слагаемые в скобках, кроме 1, с ростом k стремятся к нулю. После сокращения вынесенных за скобки множителей получим

$$\lim_{k \rightarrow \infty} \frac{y_i^{(k)}}{y_i^{(k-1)}} = \lambda_1.$$

При этом из (4) получаем

$$\lim_{k \rightarrow \infty} \frac{\mathbf{y}^{(k)}}{c_1\lambda_1^k} = \lim_{k \rightarrow \infty} \left[\mathbf{x}_1 + \left(\frac{c_2}{c_1} \frac{\lambda_2^k}{\lambda_1^k} \mathbf{x}_2 + \cdots + \frac{c_n}{c_1} \frac{\lambda_n^k}{\lambda_1^k} \mathbf{x}_n \right) \right] = \mathbf{x}_1.$$

Таким образом по мере итерирования $\mathbf{y}^{(k)}$ будет по направлению стремиться к собственному вектору \mathbf{x}_1 .

Для практической реализации этой идеи потребуется, во-первых, производить нормировку векторов на каждой итерации, чтобы их

длины не стали слишком большими или маленькими и не происходило потери значащих цифр, и, во-вторых, определить некий критерий остановки. Сформулируем алгоритм реализации степенного метода (можно встретить название «РМ-алгоритм», от англ. Power method).

Степенной метод

1. Выбрать начальный вектор $\mathbf{y}^{(0)}$ и вычислить нормированный вектор $\mathbf{z}^{(0)} = \frac{\mathbf{y}^{(0)}}{\|\mathbf{y}^{(0)}\|}$. Положить $k = 1$.
2. Вычислить следующий вектор $\mathbf{y}^{(k)} = \mathbf{A}\mathbf{z}^{(k-1)}$ и нормировать его $\mathbf{z}^{(k)} = \frac{\mathbf{y}^{(k)}}{\|\mathbf{y}^{(k)}\|}$.
3. Вычислить значения $\lambda_i^{(k)} = \frac{y_i^{(k)}}{z_i^{(k-1)}}$ для всех $i \in I$, где множество координат $I = \{i : |z_i^{(k-1)}| > \delta\}$, при некотором допуске $\delta > 0$, ниже которого координата считается нулевой.
4. Проверить сходимость: Если для каждого $i \in I$ в значениях $\lambda_i^{(k)}$ и соответствующих предыдущих значениях $\lambda_i^{(k-1)}$ совпадает необходимое число знаков, то работа алгоритма прекращается, усреднённое по всем координатам значение

$$\bar{\lambda}_1 = \left\langle \lambda_i^{(k)} \right\rangle = \frac{1}{|I|} \sum_{i \in I} \lambda_i^{(k)},$$

выбирается в качестве приближения к искомому собственному числу $\lambda_1 \approx \bar{\lambda}_1$, а соответствующий искомый собственный вектор $\mathbf{x}_1 \approx \mathbf{z}^{(k)}$. Иначе переходим на Шаг 2 с увеличением k .

Замечание 2. С использованием обозначения $\boldsymbol{\lambda}^{(k)} = \{\lambda_i^{(k)}\}_{i \in I}$ для вектора вычисленных по координатных приближений к λ_1 и нормы $\|\mathbf{x}\| = \|\mathbf{x}\|_\infty^I = \max_{i \in I} |x_i|$, можно записать условие сходимости как

$$\|\boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^{(k-1)}\| \leq \text{rtol} \cdot \max(\|\boldsymbol{\lambda}^{(k)}\|, \|\boldsymbol{\lambda}^{(k-1)}\|), \quad (5)$$

где rtol — требуемая относительная точность совпадения последовательных приближений (максимальное по модулю собственное число можно считать достаточно отдалённым от нуля и пользоваться толь-

ко допуском на относительную точность).

Представленный вариант степенного метода сходится со скоростью геометрической прогрессии со знаменателем $\left| \frac{\lambda_2}{\lambda_1} \right|$. Существует ряд улучшений, например, сдвиг на основе свойства 2 для ускорения сходимости, который можно применить при известных оценках $|\lambda_1|$ и $|\lambda_2|$. Можно ускорить сходимость для симметрических матриц, пользуясь ортогональностью их собственных векторов. Подробнее об этом и других методах ускорения сходимости см. в [1, 2].

Следует обратить особое внимание на то, что критерий остановки (5) сформулирован из общих соображений и не даёт никаких оценок погрешности приближения искомой собственной пары матрицы \mathbf{A} .

Обратные итерации

Степенной метод, как он описан выше, пригоден к нахождению *наибольшего* по модулю собственного числа. Если обратить матрицу \mathbf{A} и применить к ней степенной метод, то мы найдём наибольшее по модулю собственное число \mathbf{A}^{-1} , которое, согласно свойству 3, будет *наименьшим* по модулю собственным числом \mathbf{A} .

На шаге 2 рассмотренного выше алгоритма нужно будет находить $\mathbf{y}^{(k)} = \mathbf{A}^{-1}z^{(k-1)}$, что можно сделать без реального обращения матрицы \mathbf{A} , решая на каждом шаге систему линейных уравнений $\mathbf{A}\mathbf{y}^{(k)} = z^{(k-1)}$. При этом матрица системы \mathbf{A} не меняется от итерации к итерации, так что для неё достаточно один раз выполнить LU-разложение, а все последующие итерации будут не более трудоёмки, чем в прямом степенном методе.

Особенно полезным свойством обратного степенного метода является возможность сделать любое из собственных чисел матрицы \mathbf{A} наименьшим по модулю собственным числом матрицы $\mathbf{A} - \sigma\mathbf{E}$, *сдвинутой* на соответствующее число σ (см. свойство 2). Например, если известно, что для некоторого m

$$|\lambda_m - \sigma| < |\lambda_j - \sigma| \quad \forall j \neq m, \quad (6)$$

то обратный степенной метод для матрицы $\mathbf{A} - \sigma\mathbf{E}$ сойдётся к числу $\lambda_m - \sigma$. При этом сходимость будет тем быстрее, чем меньше отношение $\frac{|\lambda_m - \sigma|}{\max_{j \neq m} |\lambda_j - \sigma|}$.

Последнее наблюдение позволяет дополнительно ускорить сходимость обратных итераций, если осуществлять их с переменным сдвигом, на $(k + 1)$ -й итерации полагая сдвиг равным текущему приближению к искомому λ_m . Это обеспечивает *квадратичную* сходимость метода.

Обратный степенной метод со сдвигами

1. Выбрать начальный вектор $\mathbf{y}^{(0)}$ и начальный сдвиг σ_0 , удовлетворяющий (6) для искомого λ_m . Вычислить нормированный вектор $\mathbf{z}^{(0)} = \frac{\mathbf{y}^{(0)}}{\|\mathbf{y}^{(0)}\|}$. Положить $k = 1$.
2. Вычислить следующий вектор, решая $(\mathbf{A} - \sigma_{k-1}\mathbf{E})\mathbf{y}^{(k)} = \mathbf{z}^{(k-1)}$, и нормировать его $\mathbf{z}^{(k)} = \frac{\mathbf{y}^{(k)}}{\|\mathbf{y}^{(k)}\|}$.
3. Вычислить значения $\mu_i^{(k)} = \frac{z_i^{(k-1)}}{y_i^{(k)}}$ для всех $i \in I$, где множество координат $I = \{i : |y_i^{(k)}| > \delta\}$, при некотором допуске $\delta > 0$, ниже которого координата считается нулевой. Вычислить усреднённое по всем координатам приближение к λ_m

$$\sigma_k = \sigma_{k-1} + \left\langle \mu_i^{(k)} \right\rangle = \sigma_{k-1} + \frac{1}{|I|} \sum_{i \in I} \mu_i^{(k)}.$$

4. Проверить сходимость σ_k и $\mathbf{z}^{(k)}$. Если она достигнута, то на выходе полагаем $\lambda_m \approx \sigma_k$, а соответствующий искомый собственный вектор $\mathbf{x}_m \approx \mathbf{z}^{(k)}$. В противном случае переходим на Шаг 2 с увеличением k .

Замечание 3. На шаге 3 находятся величины $\mu_i^{(k)}$ по своей форме обратные величинам $\lambda_i^{(k)}$, которые находятся на шаге 3 степенного метода. Это объясняется тем, что обратные степенной метод равносильны прямому методу с матрицей $(\mathbf{A} - \sigma_{k-1}\mathbf{E})^{-1}$, но интересуют нас собственные числа необращённой матрицы $(\mathbf{A} - \sigma_{k-1}\mathbf{E})$.

Замечание 4. Обратим внимание, что в обратном методе с переменными сдвигами на каждой итерации решается система с новой матрицей и на совершение одной итерации требуется больше вре-

мени, чем при фиксированном сдвиге. В то же время общее число итераций сильно уменьшается. Здесь можно провести аналогию с методом Ньютона (квадратичная сходимость, на каждой итерации решается СЛАУ с новой матрицей) и его модифицированным, или упрощённым, вариантом (линейная сходимость, на всех итерациях СЛАУ с одной и той же матрицей). В обоих случаях конкретный баланс между числом и скоростью выполнения итераций зависит от задачи.

Так же как и для степенного метода, для симметрических матриц можно получить дополнительное ускорение метода обратных итераций с переменными сдвигами и получить *кубическую* сходимость, если в качестве приближения к σ_k брать отношение Рэлея для вектора $\mathbf{z}^{(k)}$ [1, 2].

В силу сложности определения подходящих начальных сдвигов и малой пригодности рассмотренных методов для нахождения комплексных пар собственных чисел, обычно полная задача собственных значений решается матричными преобразованиями матриц к подобным им диагональным (при возможности) или треугольным матрицам. После нахождения с достаточной точностью всех собственных чисел при необходимости можно применить обратный степенной метод для дальнейшего уточнения и нахождения собственных векторов. Рассмотрим наиболее популярный **QR**-алгоритм.

QR-алгоритм

LU- и QR-алгоритмы

Самым простым методом приведения несимметрической матрицы \mathbf{A} к треугольному виду является так называемый **LU**-алгоритм, основанный, как легко догадаться, на **LU**-разложении.

Пусть матрица \mathbf{A} может быть разложена в произведение нижне- и верхнетреугольных матриц $\mathbf{A} = \mathbf{L}\mathbf{U}$. Для определённости будем считать, что \mathbf{L} имеет единичную диагональ и потому невырождена. Меняя местами множители, получим другую матрицу $\mathbf{A}_1 = \mathbf{U}\mathbf{L}$. Выражая \mathbf{U} из второго равенства и подставляя в первое, получим

$$\mathbf{A} = \mathbf{L}\mathbf{A}_1\mathbf{L}^{-1},$$

что означает подобие матриц \mathbf{A} и \mathbf{A}_1 и совпадение их спектров.

Если матрица \mathbf{A}_1 в свою очередь может быть разложена как $\mathbf{A}_1 = \mathbf{L}_1\mathbf{U}_1$, то получим подобную ей и исходной матрице \mathbf{A} матрицу $\mathbf{A}_2 = \mathbf{U}_1\mathbf{L}_1$:

$$\mathbf{A} = \mathbf{L}\mathbf{L}_1\mathbf{A}_2\mathbf{L}_1^{-1}\mathbf{L}^{-1}.$$

При условии возможности продолжения этот процесс (при некоторых дополнительных ограничениях на матрицу \mathbf{A}) сойдётся к верхнетреугольной матрице, подобной исходной (если бы фиксировали диагональ \mathbf{U} , то сходимость была бы к нижнетреугольной матрице). Этот метод носит название **LU**-алгоритма. Одним из достаточных условий его сходимости является различие по модулю *всех* собственных чисел матрицы \mathbf{A} .

Следует отметить, что **LU**-алгоритм наследует вычислительные проблемы, присущие **LU**-разложению, а введение в него перестановок для снижения вычислительной погрешности весьма сложно и существенно затрудняет реализацию. В связи с этим наиболее широко используемым является построенный на основе ортогональных преобразований **QR**-алгоритм², по своей идее совпадающий с рассмотренным **LU**-алгоритмом.

Полагая $\mathbf{A}_0 = \mathbf{A}$ при $k = 0, 1, \dots$ строят разложение $\mathbf{A}_k = \mathbf{Q}_k\mathbf{R}_k$ и матрицу $\mathbf{A}_{k+1} = \mathbf{R}_k\mathbf{Q}_k$, подобную исходной. При этом \mathbf{Q}_k — ортонормированная, а \mathbf{R}_k — верхнетреугольная. Также как и в **LU**-

²Этот метод был представлен независимо друг от друга советским и российским математиком Верой Николаевной Кублановской [4] и англичанином Джоном Фрэнсисом [5, 6] в 1961 г.

алгоритме пределом последовательности \mathbf{A}_k является верхнетреугольная матрица, подобная матрице \mathbf{A} .

Замечание 5. Треугольная матрица в пределе последовательности $\{\mathbf{A}_k\}$ будет получаться, если все собственные числа \mathbf{A} различны по модулю. Если же у матрицы \mathbf{A} есть комплексная пара собственных чисел, то на диагонали в пределе будет блок 2×2 , имеющий эту комплексную пару своими собственными числами. При это сходимость будет именно для комплексной пары собственных чисел такого блока, хотя сами по себе его элементы не будут стремиться к какому-то пределу и могут меняться непредсказуемым образом от итерации к итерации.

Конкретный метод построения **QR**-разложения не играет роли, но наиболее быстрая и численно устойчивая реализация первым этапом включает в себя приведение матрицы к почти треугольной форме, так называемой форме Хессенберга, а затем применение **QR**-алгоритма в сформулированном выше виде к такой матрице.

Приведение матрицы к форме Хессенберга

*Матрицей Хессенберга*³ (верхней) называется матрица, у которой все элементы ниже первой поддиагонали равны нулю, т. е. элементы $a_{ij} = 0$ при $j > i + 1$:

$$\begin{pmatrix} * & * & * & \cdots & * & * & * \\ * & * & * & \cdots & * & * & * \\ 0 & * & * & \cdots & * & * & * \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & * & * & * \\ 0 & 0 & 0 & \cdots & * & * & * \\ 0 & 0 & 0 & \cdots & 0 & * & * \end{pmatrix}$$

Легко показать, что матрица **RQ**, полученная перемножением матриц **Q** и **R** ортогонального разложения Хессенберговой матрицы, останется матрицей Хессенберга (задание на самостоятельную проверку). Само же **QR**-разложение для матрицы Хессенберга выполняется за $n - 1$ *вращения Гивенса* (или аналогичные по затратам неполные *отражения*, обнуляющие всего один элемент столбца), а

³ *Карл Адольф Хёссенберг* (нем. *Karl Adolf Hessenberg*) (1904–1959), немецкий инженер-электротехник и математик.

значит занимает квадратичное время, в то время как **QR**-разложение полной матрицы требует $O(n^3)$ арифметических операций.

В связи с этим, изначальное приведение матрицы к форме Хессенберга является обоснованным шагом, существенно ускоряющим последующие итерации. Это приведение (как и само **QR**-разложение) может быть выполнено и с помощью вращений Гивенса, но чаще используют матрицы отражений. Одной из причин является лучшее распараллеливание такого процесса.

Построение матриц отражений происходит практически так же, как для триангулизации матрицы, но со сдвигом на один элемент. Так на первом шаге приведения матрицы

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix}$$

обнулим её первый столбец под элементом a_{21} умножением *слева* на матрицу

$$\mathbf{H}_1 = \mathbf{E} - 2\mathbf{v}\mathbf{v}^T, \quad \mathbf{v} = \mu_1(0, a_{21} - s_1, a_{31}, \dots, a_{n1})^T,$$

$$s_1 = \operatorname{sgn}_+(a_{21}) \sqrt{\sum_{i=2}^n a_{i1}^2}, \quad \mu_1 = \frac{1}{\sqrt{2s_1(s_1 - a_{21})}},$$

где $\operatorname{sgn}_+ x = \begin{cases} 1, & x \geq 0, \\ -1, & x < 0. \end{cases}$

В силу того, что у матрицы \mathbf{H}_1 первый столбец нулевой, умножение на $\mathbf{H}_1^{-1} = \mathbf{H}_1$ *справа* не меняет первый столбец матрицы $\mathbf{H}_1\mathbf{A}$. Таким образом, у матрицы $\mathbf{H}_1\mathbf{A}\mathbf{H}_1$, подобной исходной матрице \mathbf{A} , в первом столбце все элементы, начиная с третьего, обнуляются.

Последующие матрицы $\mathbf{H}_2, \dots, \mathbf{H}_{n-1}$ строятся аналогично.

Сдвиги и понижение размерности

Рассмотрим применение **QR**-алгоритма к матрице Хессенберга

$$\mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & b_{13} & \cdots & b_{1,n-1} & b_{1n} \\ b_{21} & b_{22} & b_{23} & \cdots & b_{2,n-1} & b_{2n} \\ 0 & b_{32} & b_{33} & \cdots & b_{3,n-1} & b_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & b_{n-1,n-1} & b_{n-1,n} \\ 0 & 0 & 0 & \cdots & b_{n,n-1} & b_{nn} \end{pmatrix}.$$

После каждой итерации будем получать матрицу \mathbf{B}_k с элементами $\{b_{ij}^{(k)}\}$.

В сформулированном выше виде **QR**-алгоритм сходится достаточно медленно (пусть мы и ускорили отдельные итерации, работая с формой Хессенберга). Если у матрицы \mathbf{B} собственные числа строго упорядочены по модулю $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$, то она сходится к треугольной матрице

$$\mathbf{U} = \begin{pmatrix} \lambda_1 & u_{12} & \cdots & u_{1n} \\ 0 & \lambda_2 & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix},$$

причём поддиагональные элементы стремятся к нулю со скоростью геометрической прогрессии

$$b_{i+1,i}^{(k)} = O\left(\left(\frac{|\lambda_{i+1}|}{|\lambda_i|}\right)^k\right).$$

Для ускорения сходимости в можно применить ту же идею сдвигов, что изучалась в обратном степенном методе. Так как последовательность $\{b_{nn}^{(k)}\}$ стремится к λ_n , то если на каждой итерации выполнять сдвиг матрицы \mathbf{B}_k на величину $b_{nn}^{(k)}$, то будет наблюдаться *квадратичная* сходимость к нулю последовательности $b_{n,n-1}^{(k)}$.

Учитывать сдвиги можно двумя способами: либо суммировать все совершённые сдвиги (понимая, что чем ближе полный сдвиг к точному собственному числу, тем ближе правый нижний элемент сдвинутой матрицы к нулю), либо каждый раз производя обратный сдвиг, то есть, выполняя итерацию **QR**-алгоритма по формулам:

$$\mathbf{B}_k - b_{nn}^{(k)}\mathbf{E} = \mathbf{Q}_k\mathbf{R}_k, \quad \mathbf{B}_{k+1} = \mathbf{R}_k\mathbf{Q}_k + b_{nn}^{(k)}\mathbf{E}.$$

Вместе со сдвигами применяют так называемую *процедуру исчерпывания*, суть которой заключается в том, что при получении достаточно точного приближения к λ_n на позиции $b_{nn}^{(k)}$ соседний поддиагональный элемент будет практически нулевым. Тогда можно отбросить последнюю строку и столбец, так как они больше не влияют на собственные числа главного минора меньшей размерности. К меньшей матрице теперь можно применять сдвиги, ускоряющие сходимость к следующему по величине собственному числу. При этом, можно просто контролировать величину поддиагонального элемента $b_{n,n-1}^{(k)}$ и считать его нулевым по достижении некоторого порога ε , но, как правило, одновременно контролируют и сходимость $b_{nn}^{(k)}$, например, считая приближение достаточно точным при выполнении условия $|b_{nn}^{(k)} - b_{nn}^{(k-1)}| < \frac{1}{3}|b_{nn}^{(k-1)}|$.

Замечание 6. Если у матрицы \mathbf{A} есть кратные действительные собственные числа, то описанный выше алгоритм найдёт их одно за другим. Если же существует комплексная пара, то при нахождении блока 2×2 в нижнем правом углу необходимо производить поочерёдные сдвиги на каждый из элементов комплексно-сопряжённой пары, что потребует действий с комплексными числами и снизит эффективность от применения сдвигов. Однако после двух сдвигов на элементы комплексно-сопряжённой пары, матрица снова становится действительной. Это позволяет выполнять переход сразу от \mathbf{V}_k к \mathbf{V}_{k+2} , оставаясь в рамках действительной арифметики. Подробно об особенностях реализации вычислений комплексно-сопряжённых пар собственных значений **QR**-алгоритмом см. в [2].

Задание для самостоятельного выполнения

Теоретические задачи

1. Обоснуйте, почему использовать отношения Рэлея в качестве приближения к λ_m в обратном степенном методе можно лишь для симметрических матриц.
2. Покажите, что форма Хессенберга сохраняется при совершении итераций **QR**-алгоритма.

Задания для программной реализации

На вход программы подаётся размерность задачи n . Сгенерируйте случайную диагональную $[n \times n]$ -матрицу Λ . Сгенерируйте случайную матрицу \mathbf{C} той же размерности. Матрица $\mathbf{A} = \mathbf{C}^{-1}\Lambda\mathbf{C}$ будет рабочей матрицей с действительными собственными числами, которые известны по матрице Λ .

1. Реализуйте степенной метод для нахождения наибольшего по модулю собственного числа матрицы \mathbf{A} и соответствующего собственного вектора.
2. Найдите все собственные пары матрицы \mathbf{A} , используя обратный степенной метод с постоянными сдвигами (выберите какие-нибудь начальные приближения к каждому из собственных чисел). Сделайте то же с переменными сдвигами, сравните число итераций и получаемую точность при использовании одинакового критерия сходимости.
3. Реализуйте **QR**-алгоритм со сдвигами и понижением размерности для нахождения всех собственных чисел матрицы \mathbf{A} . Предварительно приведите её к форме Хессенберга. Уточните каждое найденное собственное число, выполнив обратные итерации (чтобы убедиться в достаточной точности результатов **QR**-алгоритма, может хватить и одной).
4. Видоизмените исходную матрицу Λ , поменяв (если нужно) знак λ_n так, чтобы $\lambda_{n-1}\lambda_n < 0$, и поменяв в ней последние две строки местами. Постройте матрицу \mathbf{A} так же как раньше. Примените **QR**-алгоритм *без сдвигов и понижения размерности* к новой матрице \mathbf{A} . Проследите за появлением блока $[2 \times 2]$, не

сходящегося поэлементно. Доведите алгоритм до сходимости к нулю с точностью ε всех поддиагональных элементов, кроме принадлежащего этому блоку. После этого совершите несколько итераций **QR**-алгоритма, находя на каждой собственные числа этого блока. Убедитесь в их сходимости.

Возьмите следующие величины в качестве допусков:

- Допуск на абсолютную величину учитываемых при усреднении в степенном методе координат векторов $\delta = 10^{-8}$;
- Допуск на относительную погрешность определения собственных чисел в степенном методе $\text{rtol} = 10^{-6}$;
- Допуск на малость поддиагонального элемента **QR**-алгоритма для процедуры исчерпывания $\varepsilon = 10^{-8}$.

Литература

1. Вержбицкий В. М. Основы численных методов: учебник для вузов — М.: Директ-Медиа, 2013. 847 с.
2. Вержбицкий В. М. Вычислительная линейная алгебра: Учеб. пособие для вузов — М.: Высш. шк., 2009. 351 с.
3. Голуб Дж., Ван Лоун Ч. Матричные вычисления: Пер. с англ. — М.: Мир, 1999. 548 с.
4. Кублановская В. Н. О некоторых алгоритмах для решения полной проблемы собственных значений // Ж. вычисл. матем. и матем. физ., 1961, 1(4), 555–570.
5. Francis J.G.F. The QR Transformation A Unitary Analogue to the LR Transformation — Part 1 // The Computer Journal, 1961, 4(3), 265–271.
6. Francis J.G.F. The QR Transformation — Part 2 // The Computer Journal, 1962, 4(4), 332–345.