



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ _____ Фундаментальные науки

КАФЕДРА _____ Прикладная математика

Отчёт по лабораторной работе №4

Методы решения проблемы собственных значений

Студент: _____
ФН2-52Б
(Группа)

(Подпись, дата)

А. И. Токарев

(И. О. Фамилия)

(Подпись, дата)

Ю. А. Сафронов

(И. О. Фамилия)

Проверил:

(Подпись, дата)

(И. О. Фамилия)

2021 г.

Оглавление

1. Краткое описание алгоритмов	3
1.1. Метод QR разложения	3
1.2. Метод обратных итераций	4
2. Исходные данные	5
3. Результаты расчетов	6
4. Контрольные вопросы	6

1. Краткое описание алгоритмов

1.1. Метод QR разложения

Один из способов нахождения собственных значений квадратной матрицы — приведение данной матрицы к треугольному виду преобразованием подобия

$$R = P^{-1}AP,$$

где P —невырожденная матрица, которую можно найти, используя QR —алгоритм. Рассмотрим метод QR —разложения или алгоритмом Френсиса-Кублановской.

На первой итерации строится QR —разложение матрицы $A^{(0)} = A$:

$$A^{(0)} = Q_1 R_1, \quad R_1 = Q_1^{-1} A^{(0)}.$$

Затем вычислим матрицу $A^{(1)} = R_1 Q_1$ или $A^{(1)} = Q_1^{-1} A^{(0)} Q_1$.

Видим, что $A^{(0)}$ и $A^{(1)}$ подобны и имеют один и тот же набор собственных значений. На второй итерации найдём QR —разложение матрицы $A^{(1)}$ и вычисляется $A^{(2)}$. На $(k+1)$ —ой итерации определим разложение $A^{(k)} = Q_{k+1} R_{k+1}$ и построим матрицу

$$A^{(k+1)} = R_{(k+1)} Q_{(k+1)} = Q_{(k+1)}^{-1} A^{(k)} Q_{k+1}.$$

Получим последовательность матриц $\{A^{(k)}\}$, в том случае, если собственные значения A вещественны и различны по модулю, т.е. $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$, сходится к верхнетреугольной матрице. Отметим, что элементы $a_{ij}^{(k)}$ матриц $A^{(k)}$, стоящие ниже главной диагонали, сходятся к нулю со скоростью геом-ой прогрессии, т.е.:

$$|a_{ij}^{(k)}| \leq \left| \frac{\lambda_i}{\lambda_j} \right| \cdot |a_{ij}^{(k-1)}|, \quad i > j, \quad k = 1, 2, \dots$$

Отметим, что среди С.Ч матрицы A есть близкие величины, то есть

$$\left| \frac{\lambda_i}{\lambda_j} \right| \approx 1,$$

то сходимость будет очень медленной. Поэтому используют алгоритм со сдвигами: ищем собственные значения матрицы $\tilde{A} = A - \sigma E$, которые равны $\tilde{\lambda}_i = \lambda_i - \sigma$. В таком случае скорость сходимости QR —алгоритма определяется величиной

$$\left| \frac{\tilde{\lambda}_i}{\tilde{\lambda}_j} \right| = \left| \frac{\lambda_i - \sigma}{\lambda_j - \sigma} \right|.$$

1.2. Метод обратных итераций

Если известно собственное значение λ_i матрицы A или точное приближение λ_i^* , то можно рассмотреть задачу нахождения собственного вектора, отвечающему данному собственному значению.

Собственный вектор e_i ищем как нетривиальное решение системы линейных алгебраических уравнений

$$(A - \lambda_i E)e_i = 0$$

с вырожденной матрицей $(A - \lambda_i E)$. Когда λ_i известно приближенно, тогда нужно решать систему

$$(A - \lambda_i^* E)e_i,$$

решение которой может быть только тривиальным, так как матрица $(A - \lambda_i^* E)$ невырождена. Поэтому численное решение данной системы не даёт возможности вычислить соответствующий собственный вектор.

Тогда рассмотрим **метод простых итераций**. Каждая итерация данного метода состоит из двух этапов:

1. На первом этапе решается система

$$(A - \lambda_i^* E)y^{(k+1)} = x^{(k)}$$

относительно неизвестного вектора y^{k+1} .

2. На втором этапе производится нормировка решения:

$$x^{(k+1)} = \frac{y^{(k+1)}}{\|y^{(k+1)}\|}.$$

В качестве $x^{(0)}$ можно взять любой нормированный вектор. При условии, что известное приближение λ_i^* достаточно близко к истинному значению λ_i , последовательность векторов x^k быстро сходится к собственному вектору e_i , соответствующему собственному значению λ_i .

2. Исходные данные

Даны матрицы ($dim = 4$):

$$A_{20} = \begin{pmatrix} 99.4000 & -2.9000 & -9.9800 & 0.6300 \\ -2.9000 & 106.4000 & -9.4300 & -8.0200 \\ -9.9800 & -9.4300 & -159.4000 & -5.8900 \\ 0.6300 & -8.0200 & -5.8900 & 58.2000 \end{pmatrix}$$

$$A_{23} = \begin{pmatrix} -182.2000 & 1.6100 & 7.6500 & -9.1000 \\ 1.6100 & -43.4000 & 5.5400 & -6.8500 \\ 7.6500 & 5.5400 & 12.6000 & 9.3200 \\ -9.1000 & -6.8500 & 9.3200 & -77.2000 \end{pmatrix}$$

3. Результаты расчетов

4. Контрольные вопросы

1. Почему нельзя находить собственные числа матрицы A , прямо решая уравнение $\det(A - \lambda E) = 0$, а собственные векторы — «по определению», решая систему $(A - \lambda_i E)e_i = 0$?

Для нахождения собственных чисел матрица, нам надо составить характеристический многочлен и найти его корни, что для многочленов высокой степени является трудоемким процессом. Данный подход становится неудовлетворительным, если речь идёт о вычислении собственных значений матриц, имеющих порядок m в несколько десятков (или даже сотен). Одна из причин состоит в том, что $Ax = \lambda x$ и $\lambda^m + p_1\lambda^{m-1} + p_2\lambda^{m-2} + \dots + p_{m-1}\lambda + p_m = 0$ формально эквивалентны, они имеют разную обусловленность. Так как корни многочлена $P_m(x)$ высокой степени срезвычайно чувствительны к погрешностям в коэффициентах, то на этапе вычисления коэффициентов характеристического уравнения может быть в значительной степени потеряна информация о собственных значениях матрицы. Если исходить непосредственно из определения собственного вектора, то e_i следует искать как нетривиальное решение системы линейных алгебраических уравнений

$$(A - \lambda_i E)e_i = 0$$

с вырожденной матрицей $(A - \lambda_i E)$. Но обычно λ_i известно лишь приближенно, и в действительности приходится решать систему

$$(A - \lambda_i^* E)e_i = 0,$$

где λ_i^* — достаточно точное приближение к собственному значению λ_i . Решение данной системы быть только тривиальным, так как матрица $(A - \lambda_i^* E)$ невырождена. Поэтому непосредственное численное решение не дает возможности вычислить соответствующий собственный вектор.

2. Докажите, что ортогональное преобразование подобия сохраняет симметрию матрицы.

Ортогональное преобразование подобия имеет вид:

$$R = P^{-1} A P,$$

где $P^{-1} = P^T$, $A = A^T$. Тогда

$$\begin{aligned} R^T &= (P^{-1} A P)^T = P^T (A P^{-1})^T = P^T A^T (P^{-1})^T = P^{-1} A P = R, \\ &\implies R^T = R. \end{aligned}$$

3. Как преобразование подобия меняет собственные векторы матрицы?

Можно рассматривать матрицу P как матрицу перехода. $B = P^{-1} A P$ матрицы A и B подобны.

Полученная в результате преобразования подобия матрица имеет тот же набор собственных чисел:

$$\begin{aligned} \det(P^{-1} A P - \lambda E) &= \det(P^{-1} (A - \lambda E) P) = \\ &= \det(P^{-1}) \det(A - \lambda E) \det(P) = \det(A - \lambda E). \end{aligned}$$

Таким образом, характеристические многочлены и собственные числа матриц A и $P^{-1} A P$ совпадают. Соответствующие собственные векторы x и x' не совпадают, но, т.к. $P^{-1} A P x' = \lambda x' \Rightarrow A P x' = \lambda P x'$, они связаны равенством $x = P x'$.

4. Почему на практике матрицу A подобными преобразованиями вращения приводят только к форме Хессенберга, но не к треугольному виду?

Рассматривая алгоритм подобных преобразований вращения, заметим, что обнуляются все элементы, лежащие левее элемента стоящего на поддиагонали, то построив такую последовательность элементарных вращений, которая приведет матрицу A к форме Хессенберга:

$$A^* = T_{kl} A T_{kl}^{-1} = T_{kl} A T_{kl}^T$$

где T_{kl} -матрица, в которой все элементы главной диагонали равны 1, кроме элементов стоящих на пересечении k -ых столбца и строки и l -ых столбца и строки(они равны $\alpha = \cos \varphi$), а все элементы вне главной диагонали равны 0, за исключением элемента на пересечении k -ого столбца и l -ой строки(он равен $-\beta = -\sin \varphi$) и элемента стоящего на пересечении l -ого столбца и k -ой строки(он равен $\beta = \sin \varphi$), а A^* отличается от матрицы A лишь двумя

строками и двумя столбцами с номерами k, l , при этом в матрице A^* элемент $a_{l,k-1}^* = 0$.

По построению матриц T_{kl} видно, что $k > l$ и $k > 1$, а значит можем сделать вывод, что не возможно данными преобразованиями занулить поддиагональные элементы.

5. Оцените количество арифметических операций, необходимое для приведения произвольной квадратной матрицы A к форме Хессенберга.

Для вычисления элементов матрицы T_{kl} требуется 5 мультипликативных операций. Необходимо обнулить все элементы ниже диагонали, примыкающей к главной в столбцах с 1 по $n - 2$. В k -ом столбце необходимо обнулить $n - k - 1$ элемент. Умножение слева и справа на матрицы T_{kl} и T_{kl}^T соответственно изменяет в матрице A $4n - 2k + 2$ элемента. Для изменения одного элемента требуется 2 мультипликативные операции. В итоге получаем:

$$\sum_{k=1}^{n-2} 5 \cdot (n - k - 1)(4n - 2k + 2) \cdot 2 = \frac{50n^3}{3} - 40n^2 + \frac{10n}{3} + 20.$$

6. Сойдется ли алгоритм обратных итераций, если в качестве начального приближения взять собственный вектор, соответствующий другому собственному значению? Что будет в этой ситуации в методе обратной итерации, использующем отношение Рэлея?

В качестве начального приближения в методе обратных итераций можно взять любой нормированный вектор. Пусть e_i , $i = \overline{1, n}$ — ОНБ из собственных векторов матрицы A .

$$\begin{aligned} (A - \lambda_j^* E)y &= x; \\ y &= \sum_{i=1}^n \alpha_i e_i, \quad x = \sum_{i=1}^n c_i e_i; \\ \sum_{i=1}^n \alpha_i (\lambda_i - \lambda_j^*) &= \sum_{i=1}^n c_i e_i; \\ \alpha_i &= \frac{c_i}{\lambda_i - \lambda_j^*}; \\ y &= \sum_{i=1}^n \frac{c_i}{\lambda_i - \lambda_j^*} e_i = \frac{1}{\lambda_i - \lambda_j^*} \left(c_j e_j + \sum_{i \neq j} \frac{\lambda_j - \lambda_j^*}{\lambda_i - \lambda_j^*} c_i e_i \right). \end{aligned}$$

Если в качестве начального приближения взять собственный вектор, соответствующий другому собственному числу:

$$y = \frac{1}{\lambda_i - \lambda_j^*} \left(c_j e_j + \frac{\lambda_j - \lambda_j^*}{\lambda_k - \lambda_j^*} c_k e_k \right).$$

Если $|\lambda_j - \lambda_j^*| \ll |\lambda_k - \lambda_j^*|$, то второе слагаемое правой части мало по сравнению с первым. Следовательно алгоритм сойдется к e_j .

Если в методе обратных итераций использовать отношение Рэлея, а в качестве начального приближения $x^{(0)}$ выбрать собственный вектор e_k , соответствующий другому собственному значению, то метод сойдется к собственному числу, соответствующему собственному вектору e_k .

7. Сформулируйте и обоснуйте критерий останова для QR -алгоритма отыскания собственных значений матрицы.

Так как последовательность матриц A_k сходится к верхнетреугольной матрице R , на главной диагонали которой стоят собственные значения, то используя тот факт, что QR -алгоритм последовательно обнуляет элементы начиная с $a_{n,1}$ до $a_{n,n-1}$, то итерационный метод поиска собственного значения следует продолжать пока не будет выполняться неравенство $|a_{n,n-1}| < \varepsilon$, затем считая что $\lambda_i = a_{n,n}$ переходить к задаче меньшей размерности т.е. искать спектр матрицы размерности $(n-1) \times (n-1)$.

8. Предложите возможные варианты условий перехода к алгоритму со сдвигами. Предложите алгоритм выбора величины сдвига.

При помощи леммы Гершгорина можем оценить диапазон собственных значений и в случае, если оценка диапазона меньше единицы, то мы получим достаточное условие того, что отношение собственных значений близко к единице, и можно будет перейти к алгоритму со сдвигами, взяв в качестве величины сдвига среднее значение из оценки диапазона.

Так же если элементы $a_{ij}^{(k)}$ матриц $A^{(k)}$, стоящие ниже главной диагонали $\frac{a_{ij}^{(k)}}{a_{ij}^{(k-1)}} \leq \left| \frac{\lambda_i}{\lambda_j} \right| \approx 1, i > j, k = 1, 2, \dots$, то алгоритм будет сходиться очень медленно и следует переходить к алгоритму со сдвигами. В качестве величины сдвига можно взять $a_{n,n}^{(k)}$.

9. Для чего нужно на каждой итерации нормировать приближение к собственному вектору?

Если $|\lambda| > 1$, то последовательность норм векторов стремится к бесконечности, если $|\lambda| < 1$, то последовательность норм векторов стремится к нулю и возможно исчезновение порядка. Для предупреждения этих ситуаций вектор x^k нормируют. То есть приближение к собственному вектору необходимо нормировать для того чтобы застраховаться от накопления погрешностей и выхода значений переменных за пределы типа.

10. Приведите примеры использования собственных чисел и собственных векторов в численных методах.

- 1) С помощью собственных чисел можно сделать вывод о числе обусловленности матрицы.
- 2) В электрических и механических системах собственные числа отвечают собственным частотам колебаний, а собственные векторы характеризуют соответствующие формы колебаний.
- 3) Одна из задач, которая дает геометрическую интерпретацию собственных векторов, есть приведение кривых второго порядка к каноническому виду. Собственные вектора образуют главные направления кривых второго порядка.