



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

ОТЧЕТ
К ЛАБОРАТОРНОЙ РАБОТЕ
НА ТЕМУ:

*Итерационные методы решения систем
линейных алгебраических уравнений
Вариант 1*

Студент	ФН2-51Б (Группа)	_____ (Подпись, дата)	Н. О. Акиншин (И. О. Фамилия)
Студент	ФН2-51Б (Группа)	_____ (Подпись, дата)	А. С. Джагарян (И. О. Фамилия)

Оглавление

1. Контрольные вопросы	3
----------------------------------	---

1. 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 $(A - \lambda E)e_j = 0$?

Ответ. Исходя из уравнения $\det(A - \lambda E) = 0$, матрица $A - \lambda E$ должна быть вырожденной, однако в силу того, что все измерения проводятся с некоторой погрешностью и все числа в компьютере хранятся тоже с некоторой ошибкой, то матрица $A - \lambda E$ почти никогда не будет вырожденной. Также стоит добавить, что нахождение определителя матриц больших размерностей задача очень трудная с вычислительной точки зрения.

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

Ответ. Пусть имеем матрицу A — симметричная т.е. $A^T = A$ и ортогональное преобразование Q т.е. $Q^T = Q^{-1}$. Тогда подобная матрица B , полученная ортогональным преобразованием т.е. $B = Q^{-1} \cdot A \cdot Q$ в силу ортогональности $B = Q^T \cdot A \cdot Q$. Транспонируем обе части и используем $A^T = A$, тогда получим $B^T = Q^T \cdot A^T \cdot Q = Q^T \cdot A \cdot Q = B$.

- 3) Как преобразование подобия меняет собственные векторы матрицы?

Ответ. В ходе преобразования

$$B = P^{-1}AP$$

собственный вектор u матрицы A , соответствующий собственному значению λ , меняется следующим образом

$$v = P^{-1}u.$$

Если считать P матрицей перехода, то это означает, что вектор u переходит в другой базис, то есть растягивается или поворачивается.

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

Ответ. Заметим следующее, матрица поворота T_{kl} — ортогональна и преобразует исходную матрицу по следующему закону $A^* = T_{kl} \cdot A \cdot T_{kl}^T$. Такой поворот обращает в ноль следующий коэффициент $a_{l,k-1}^*$. Матрица T_{kl} состоит из нулей, кроме $T_{kl}^{kk}, T_{kl}^{kl}, T_{kl}^{lk}, T_{kl}^{ll}, T_{kl}^{ii} \forall i$ (верхние индексы обозначают соответствующие координаты матрицы). Если требуется обнулить коэффициенты под главной диагональю, тогда $a_{i,i-1}^* = 0 \forall i$ т.е. $l = i = k$ т.е. требуется совершить следующее преобразование $A^* = T_{ii} \cdot A \cdot T_{ii}^T$. Однако матрица T_{ii} не существует т.к. она определяется для разных индексов. Таким образом после приведение матрицы данными преобразованиями к виду Хессенберга нужно будет придумать какое-то другое преобразование обнуляющее коэффициенты под главной диагональю, что может вызвать вычислительные сложности т.е. такой алгоритм будет работать дольше чем QR разложение.

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

Ответ. Рассмотрим алгоритм приведения произвольной квадратной матрицы A к форме Хессенберга.

HessenbergDecomposition(Matrix)

1: prod=0, $\alpha = 0$, $\beta = 0$, koren=0, tmp1=0, tmp2=0

2: **for** k = 1; k < size - 1; k++ **do**

3: **for** l = k+1; l < size; l++ **do**

4: $koren = \sqrt{Matrix[l][k-1] * Matrix[l][k-1] + Matrix[k][k-1] * Matrix[k][k-1]}$

```

5:    $\alpha = \frac{Matrix[k][k-1]}{koren}$ 
6:    $\beta = \frac{Matrix[l][k-1]}{koren}$ 
7:   for index = k-1; index < size; index++ do
8:       tmp1 = alpha * Matrix[k][index] + beta * Matrix[l][index]
9:       tmp2 = alpha * Matrix[l][index] - beta * Matrix[k][index]
10:      Matrix[k][index] = tmp1
11:      Matrix[l][index] = tmp2
12:   end for
13:   for index = 0; index < size; index++ do
14:       tmp1 = alpha * Matrix[index][k] + beta * Matrix[index][l]
15:       tmp2 = alpha * Matrix[index][l] - beta * Matrix[index][k]
16:       Matrix[index][k] = tmp1
17:       Matrix[index][l] = tmp2
18:   end for
19: end for
20: end for

```

Таким образом количество умножений можно вычислить по следующей формуле

$$prod = \sum_{k=1}^{n-2} \sum_{l=k+1}^{n-1} (4 + \sum_{i=k-1}^{n-1} 4 + \sum_{i=0}^{n-1} 4) = \frac{2}{3}(5n^3 - 9n^2 - 8n + 12)$$

- 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 Рэля? Привести пример.

Ответ. Рассмотрим обычный метод обратной итерации. Пусть σ_m – приближенное число к некоторому собственному значению λ_m , x_0^j – собственный вектор, соответствующий собственному значению $\lambda_j \neq \lambda_m$. Тогда рассмотрим

$$(A - \sigma_m E)y_{k+1} = x_k.$$

Заметим

$$y_{k+1} = (A - \sigma_m E)^{-k} x_0^j$$

Для произвольного вектора x

$$(A - \sigma_m E)^{-k} x = (\lambda_m - \sigma_m)^{-k} \left(\xi_m e_m + \sum_{i=1; i \neq m}^n \left(\frac{\lambda_m - \sigma_m}{\lambda_i - \sigma_m} \right)^k \xi_i e_i \right)$$

где e_i – собственные вектора для матрицы A , соответствующие собственному значению λ_i . Выражение

$$\sum_{i=1; i \neq m}^n \left(\frac{\lambda_m - \sigma_m}{\lambda_i - \sigma_m} \right)^k \xi_i e_i \rightarrow 0, \text{ при } k \rightarrow \infty.$$

Выражение выше стремится к 0 из-за выбора собственного значения и приближения к нему, а значит к 0 сойдется и элементы любого начального приближения, соответствующих собственным векторам e_i , $i \neq m$. В силу того, что любой собственный вектор задан с некоторой машинной погрешностью, то и коэффициент ξ_m при e_m будет отличен от 0, а значит метод обратной итерации будет сходиться к собственному вектору e_m , но медленнее.

Рассмотрим метод обратной итерации, в котором использовано отношение Рэлея. Пусть $\sigma_0 \approx \lambda_m$, x_0^j – собственный вектор, соответствующий собственному значению $\lambda_j \neq \lambda_m$,

$$\sigma_1 = \frac{(Ax_0^j, x_0^j)}{(x_0^j, x_0^j)} \approx \lambda_j$$

То есть получаем, что σ_1 – приближенное значение λ_j . Тогда следующий вектор $x_1 \approx (\lambda_j - \sigma_1)^{-k} \xi_j e_j$. То есть алгоритм будет сходиться к собственному значению λ_j , а следовательно, к собственному вектору x_0^j , в силу того, что следующее приближение зависит только от начального приближения собственного вектора, а не от начального приближения собственного значения.

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

Ответ. Пусть

$$A_k = Q_k R_k,$$

считая $A_0 = A$. Тогда будем считать

$$A_{k+1} = R_k Q_k$$

Тогда критерием останова для заданного малого числа ε будем считать

$$\max_{0 \leq j < i < n} |a_{ij}| < \varepsilon$$

Тогда с заданной точностью можно утверждать, что A_{k+1} является верхнетреугольной, а следовательно собственные значения лежат на главной диагонали.

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

Ответ. Из теории известно, что элементы a_{ij}^k матриц A^k , стоящие ниже главной диагонали, сходятся к нулю со скоростью геометрической прогрессии, знаменатель которой равен модулю отношения соответствующей пары собственных значений т.е.

$$|a_{ij}^k| \leq \left| \frac{\lambda_i}{\lambda_j} \right| \cdot |a_{ij}^{k-1}|$$

Заметим, что если среди собственных чисел матрицы A есть близкие по величине, то есть для некоторых значений i и j $\frac{\lambda_i}{\lambda_j} \approx 1$, то сходимость будет очень медленной. Поэтому на практике часто используют алгоритм со сдвигами. В этом случае ищут собственные значения не матрицы A , а матрицы $A - \sigma E$, которые равны $\lambda_i - \sigma$. При таком подходе скорость сходимости QR-алгоритма определяется величиной $\frac{\lambda_i - \sigma}{\lambda_j - \sigma}$. Если σ является хорошим приближением для λ_i , то это соотношение будет много меньше единицы и алгоритм будет быстро сходиться.

Тогда, если нам известно например из теоремы о кругах Гершгорина как располагаются собственные значения, то если они близки к друг другу, то имеет смысл использовать метод сдвигов. Либо можно задать условие: если количество итераций либо количество умножение превышает некоторое пороговое значение, то начинаем использовать сдвиги. Также можно использовать условие, что если относительная величина нормы ниже треугольной части матрицы больше некоторого значения, то используем сдвиги.

В качестве σ при поиске λ_i можно выбрать например a_{ii}^k элемент на диагонали. Данное приближение будет не плохим т.к. с увеличением номера итерации элементы на диагонали матрицы приближаются к соответствующим собственным значениям.

Либо можно в качестве σ выбрать значение, вычисленное по формуле Рэлея $\sigma_m = \frac{(Ae_m, e_m)}{(e_m, e_m)}$, где e_m собственный вектор e_m можно приближенно взять соответствующий вектор столбец из матрицы $R = R_k \cdot R_{k-1} \cdot R_1$.

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

Ответ. Пусть σ_m – приближенное число к некоторому собственному значению λ_m . Тогда рассмотрим

$$(A - \sigma_m E)y_{k+1} = x_k,$$

где x_0 — некоторое начальное приближение. Тогда справедливо

$$y_{k+1} = (A - \sigma_m E)^{-1}x_k$$

Заметим, что это эквивалентно

$$y_{k+1} = (A - \sigma_m E)^{-k}x_0$$

Для произвольного вектора x

$$(A - \sigma_m E)^{-k}x = (\lambda_m - \sigma_m)^{-k} \left(\xi_m e_m + \sum_{i=1; i \neq m}^n \left(\frac{\lambda_m - \sigma_m}{\lambda_i - \sigma_m} \right)^k \xi_i e_i \right) \approx (\lambda_m - \sigma_m)^{-k} \xi_m e_m, k \gg 1,$$

где e_i – собственные вектора для матрицы A , соответствующие собственному значению λ_i .

Получается, что

$$y_{k+1} \approx (\lambda_m - \sigma_m)^{-k} \xi_m e_m.$$

Тогда заметим, что по условию σ_m достаточно хорошее приближение λ_m , значит $|\lambda_m - \sigma_m| \approx 0$, а из этого следует, что $(\lambda_m - \sigma_m)^{-k}$ достаточно большое число. Значит с каждой итерацией вектор y_k будет сходиться к собственному вектору, но в силу неотнормированности, он будет его модуль будет становиться значительно больше с каждой новой итерацией. Значит каждый раз будет происходить деление на очень маленькое число в натуральной степени, что приведет к значительным ошибкам при вычислении. Тогда получается, что необходимость нормировки приближения к собственному вектору вызвана повышением точности вычислений.

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

Ответ. Задача вычисления собственных значений возникает при: 1) Использовании метода главных компонент РСА для уменьшения размерности данных, сохраняя при этом основную информацию. 2) При решении систем дифференциальных уравнений с постоянными коэффициентами. 3) Решение уравнения Шредингера сводится к вычислению собственных значений для поиска спектра энергий волновой функции.

Таблица 1. Количество итераций в зависимости от метода поиска собственных значений

	Со сдвигом	Без сдвига
С приведением к форме Хессенберга	7	33
Без приведения к форме Хессенберга	9	33

Таблица 2. Количество операций умножения и деления в зависимости от метода поиска собственных значений

	Со сдвигом	Без сдвига
С приведением к форме Хессенберга	924	4592
Без приведения к форме Хессенберга	2124	7884