QR итерации

QR метод решает проблему нахождения всех собственных значений матрицы  одновременно. Алгоритм стартует с матрицы . На -ой итерации вычисляем QR факторизацию

 (1)

и находим следующее приближение

 (2)

которое стремится к верхней треугольной матрице с ростом .

Поскольку из (1, 2) следует

, (3)

то все матрицы  являются подобными.

Диагональные элементы матрицы  сходятся к собственным значениям матрицы . Произведение ортогональных матриц  сходится к матрице собственных векторов .

Если , то  итерации сохраняют симметрию, поэтому последовательность  сходится к матрице, которая является треугольной и симметричной, то есть диагональной. Итерационный процесс (1, 2) продолжается до тех пор, пока внедиагональные элементы матрицы  больше заданной точности . *Пример*.



.

Диагональные элементы  являются аппроксимацией собственных значений  матрицы . *Замечание*. Алгоритм (1, 2) без сдвига может не работать.

*Пример*.

 - алгоритм зависает.

*Теорема*. Пусть  Тогда все собственные значения  различные и положительные

 . (4)

Для матрицы  существует спектральное разложение

 (5)

где - верхняя треугольная матрица с неотрицательными диагональными элементами.

Тогда базовый алгоритм  последовательно находит  которые сходятся к диагональной матрице .

*Доказательство*. Для простоты будем полагать, что каждая найденная факторизация включает верхнюю треугольную матрицу  с неотрицательными диагональными элементами, что обеспечивает единственность разложения. Из (5) следует

 (6)

Из (5, 6) находим

 (7)

Рассматривая , находим

 (8)

Таким образом, заключаем, что . Отсюда следует, что . Так как разложение единственное, то  (а значит ) и .

Наконец, мы находим, что

.

Последняя сходимость имеет место в силу того, что  является симметричной и подобной . 

*Замечание*. Теорема будет выполняться, если убрать требование  и что имеет неотрицательные диагональные элементы.

Рассмотренный базовый алгоритм является затратным. Он требует выполнения на каждой итерации  операций. Кроме того, он может иметь медленную скорость сходимости, в зависимости от собственных значений . Существуют три подхода для улучшения базового алгоритма.

1. Редуцировать матрицу к подобной верхней матрице Хессенберга. При этом число операций на одну итерацию сокращается до . Кроме того,  итерации не нарушают структуру матрицы Хессенберга.
2. Найденные собственные значения можно убрать, что значительно ускоряет процесс вычислений.
3. В алгоритме можно использовать сдвиги.

Рассмотрим редукцию матрицы к подобной верхней матрице Хессенберга , в итоге обе матрицы будут иметь одинаковый набор собственных значений. С это целью удобно использовать ортогональные отражения Хаусхолдера, которые последовательно обнуляют столбцы элементов ниже первой под-диагонали.

После первого отражения находим

.

В полученной матрице можно исключить первую строку и первый столбец, а затем сделать аналогичное отражение для матрицы меньшей размерности. Повторяя эту рекурсию, в итоге получим

.

После редукции быстрый алгоритм  имеет вид



Рассмотрим процесс исчерпывания собственных значений (дефляцию). Наш алгоритм с редукцией Хессенберга тратит теперь  операций на одну итерацию, но сходится очень медленно. Обычно требуется выполнить тысячи итераций.

Заметим, что если , которая сходится к верхней треугольной, имеет форму



то  является собственным значением матриц  и в силу их подобия. Оставшийся спектр матрицы совпадает с собственными значениями матрицы . Итак, если нам повезло и элемент , то мы нашли одно приближенное значение и далее можем работать с матрицей  меньшей размерности. Этот процесс называется дефляцией.

Рассмотрим алгоритм со сдвигом.



Алгоритм итерация по прежнему сохраняет структуру Хессенберга и создает последовательность подобных матриц. Идея сдвига – ускорить сходимость  к нулю. Разумным выбором кажется сдвиг , поскольку при сходимости  будет приближением собственного значения . Но это не так- контрпример: .

Уилкинсон придумал сдвиг, который работает для симметричных матриц. Он предложил выбирать подматрицу 2x2  и находить оценку собственного значения. Пусть эта подматрица имеет вид . Тогда сдвиг находится по формуле



Если  то  произвольно.

алгоритм придумал John Francis известный английский ученый в 1961 году.

