Eigenvalue problem

Команда:

Семикрас Александр

Пшеницын Артем

Словеснов Максим

Климов Глеб

NLA team, 2023

Постановка задачи

- Задача:

  реализовать различные способы нахождения собственных значений, и **сравнить** их между собой
- Значение и применение:

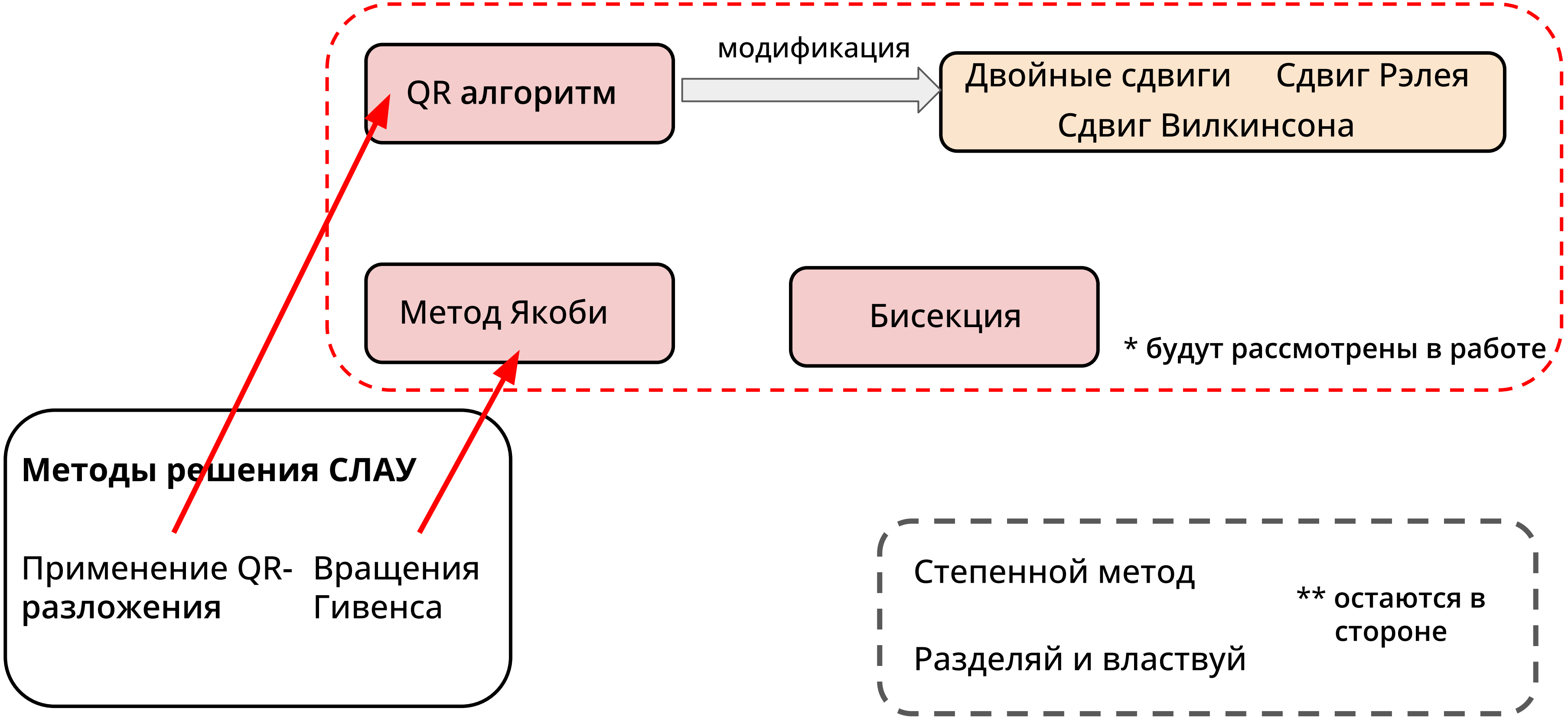
  промежуточный этап для поиска корней полиномов, решении СЛАУ и т.д.
- Мера качества:

  точность, сложность, скорость сходимости, устойчивость

Eigenvalue problem

казалось бы готовая формула

$$det(A - \lambda E) = 0$$



Особенности

- Исключительно **итеративные методы**, за конечное число шагов не справимся точно!
- Близкой задачей является нахождение собственных векторов v, соответствующих собственным числам в смысле $Av = \lambda v$. Обычно не требуется находить все собственные векторы, но они могут появляться как побочный продукт алгоритмов, находящих собственные значения.

QR-алгоритм

- Используется для решения полной задачи собственных значений, через вещественную **форму Шура**
- В ортодоксальной форме - метод простых итераций для QR-разложения.
- В практической реализации матрица приводится к верхнегессенберговой форме, что обеспечивает **кубическое** время работы
- Редукция по мере вычисления собственных значений.
- Использование **сдвигов**, для улучшения характера сходимости
- Проблема сходимости QR - алгоритма - **открытый вопрос**

$$A_k = Q_k R_k \to A_{k+1} = R_k Q_k = Q_k^* A_k Q_k$$

QR-алгоритм: проблемы со сходимостью

Достаточное условие сходимости алгоритма:

$$A = X^{-1}\Lambda X, \Lambda = \begin{bmatrix} \Lambda_{m,m} & 0 \\ 0 & \Lambda_{n-m,n-m} \end{bmatrix} \implies a^k_{m+1,m} = O(\left|\frac{\lambda_{m+1}}{\lambda_m}\right|^k)$$

Условие гарантирует **линейную скорость** сходимости. Пускай теперь есть оценка собственного значения вида $\hat{\lambda}_{m+1} = \lambda_{m+1} + o(1)$

Переходя к матрице $A - \hat{\lambda}_{m+1} I$:

$$a^k_{m+1,m} = O(\left|\frac{o(1)}{\lambda_m - \hat{\lambda}_{m+1}}\right|^k)$$

QR-алгоритм: использование сдвигов

- **Сдвиг Релея** сдвиг на значение диагонального элемента
- **Сдвиг Вилкинсона** сдвиг на вычисленное собственное значение малой матрицы

- Проблема:

  комплексные собственные значения! Вещественные оценки не могут улучшить характер сходимости
- Решение проблемы:

  использование **неявных сдвигов**

QR-алгоритм: неявная Q-теорема

- **Формулировка**:

  *Пусть $Q^T AQ = H$ - неразложимая верхняя матрица **Гессенберга**. Тогда первый столбец матрицы **Q** однозначно (с точностью до знаков) определяет её **2**-й … **n**-ый столбцы.*

- **Следствие:**

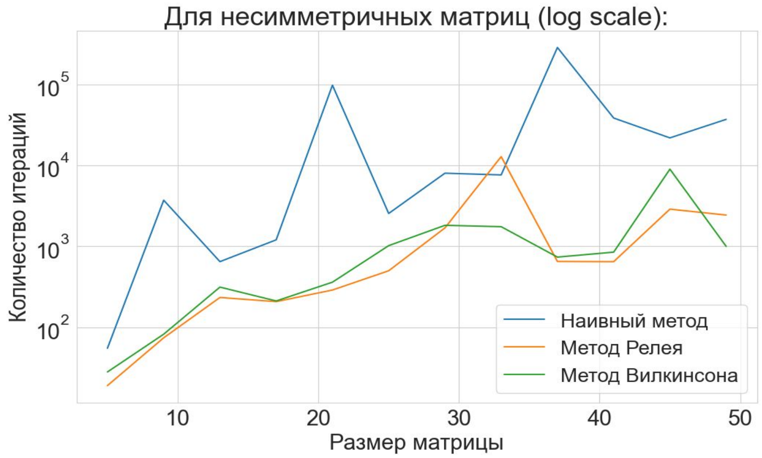
  Чтобы в QR-алгоритме вычислить по A_i матрицу $A_{i+1} = Q_i^T A_i Q_i$ достаточно:

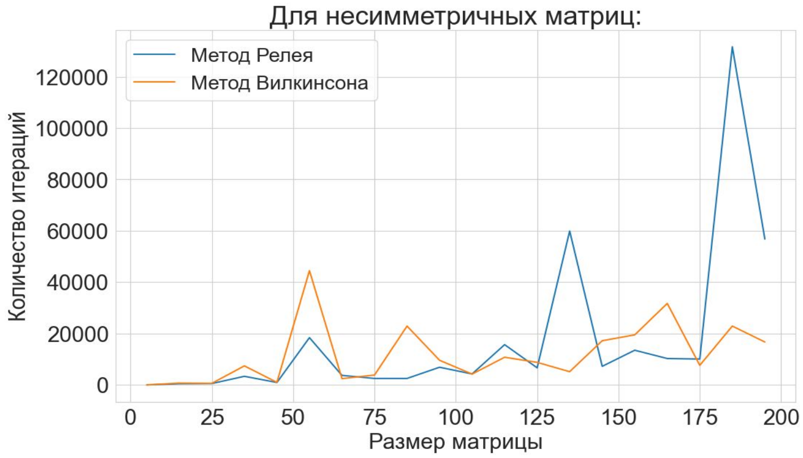
1. вычислить первый столбец матрицы Q_i(который параллелен первому столбцу матрицы $A_i - \sigma_i I$ и может быть получен из него лишь нормировкой) и
2. выбрать остальные столбцы матрицы Q_i так, чтобы Q_i была ортогональна, а A_{i+1} была неразложимой гессенберговой матрицей.

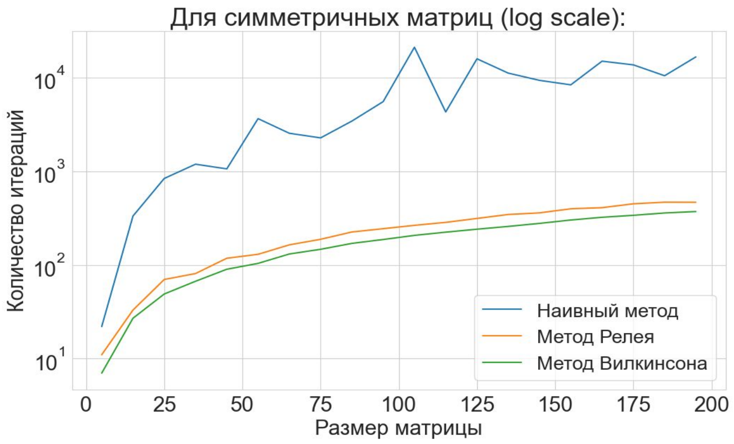
Francis Implicit Double Shift

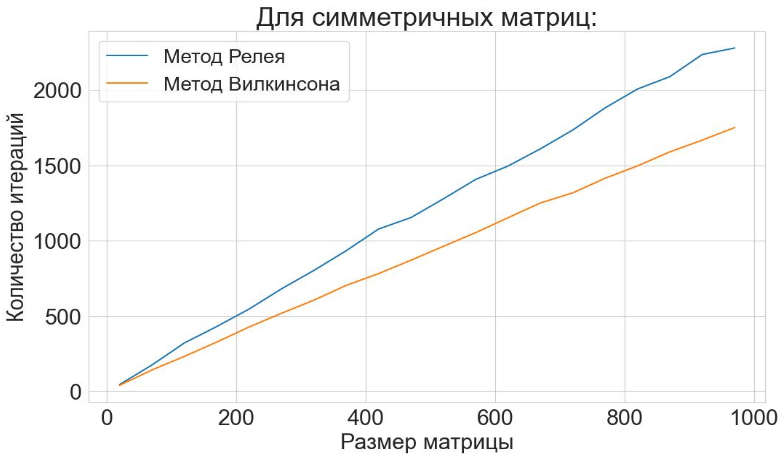
- В результате двух комплексно-сопряженных сдвигов, получаем вещественную матрицу $M = (H - \hat{\lambda} I)(H - \lambda I) = H^2 - H(\hat{\lambda} + \lambda) + |\lambda|^2 I$
- Матрица приведения M в верхнегессеберговую форму - Q - зануляет нижние элементы первого столбца M.
- По теореме о единственности, восстанавливаем матрицу перехода по ее первому столбцу за $O(n^2)$
- Ничего не изменится, если $\lambda, \hat{\lambda}$ - вещественные!

$$QHQ^* = \begin{bmatrix} * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ 0 & 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & * & * \end{bmatrix} \xrightarrow[Q']{} \begin{bmatrix} * & * & * & * & * & * & * \\ * & * & * & * & * & * & * \\ 0 & * & * & * & * & * & * \\ 0 & 0 & * & * & * & * & * \\ 0 & 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & 0 & * & * \end{bmatrix}$$









- ➔ Простые сдвиги **увеличивают сходимость** на порядок
- ➔ Для **симметричных** матриц обеспечивают **устойчивость по итерациям**

Метод бисекций:

- Используется для решения **полной задачи** собственных значений для симметричных матриц
- В реализации используется приведение симметричной матрицы к форме Хессенберга
- В основе метода лежит разложение вида

$$A - \sigma E = U^T D(\sigma) U$$

где ***A*** - исходная трехдиагональная матрица, ***D*** - диагональная матрица

Jacobi eigenvalue algorithm

➔Не нужно приведение к трехдиагональной форме
➔Медленный, но точный (даже для маленьких с.з.)
➔с.в. получаются инплейс без потерь в скорости

Идея:

Вращениями занулять недиагональные элементы, вплоть до диагонализации:

$$D_0 = A$$
$$D_{i+1} = J_i^T\, D_i\, J_i$$
$$D_i \to D - \text{диагональная}$$

$P_i(X) = J_i^T\, X\, J_i$ - зануляет какие-то Xjk и Xkj

J_i - унитарная матрица Гивенса

Все операции **линейной** сложности!

В итоге: $D \approx U^T A U$, D - с.з, U - с.в.

Jacobi eigenvalue algorithm

Как выбрать элемент для зануления на каждой итерации?

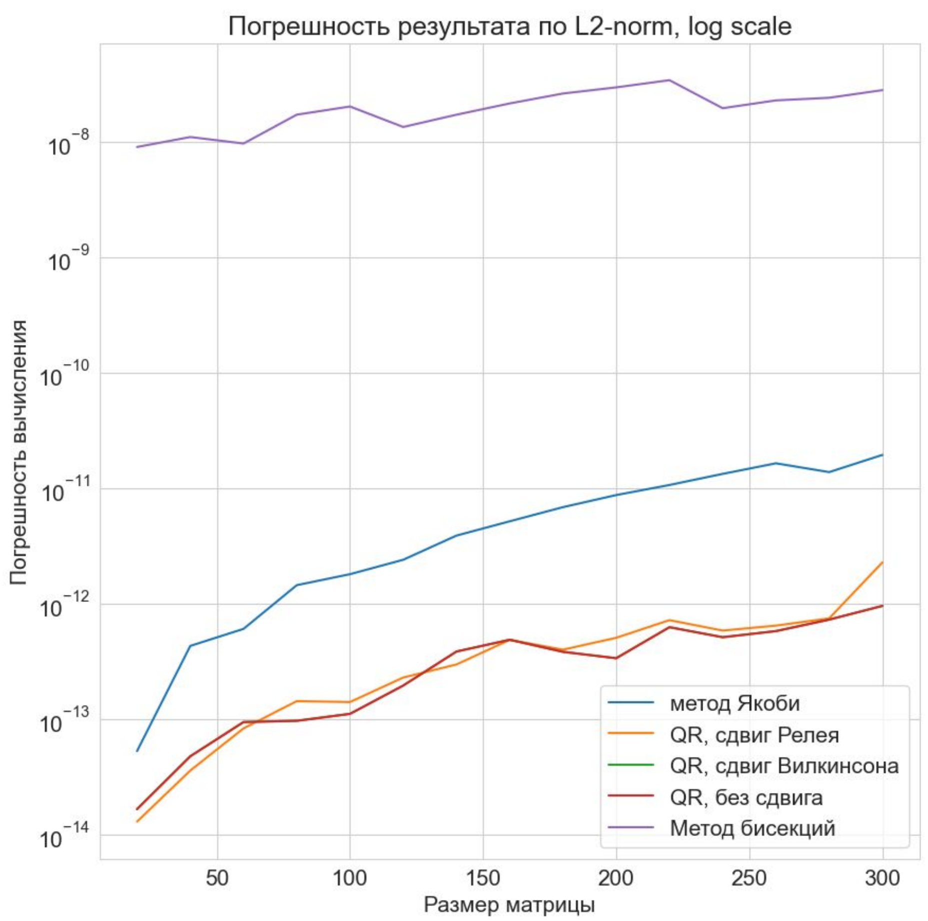
$$off(A) = \sqrt{\sum_{k \neq l}^{n} A_{kl}} \rightarrow 0$$

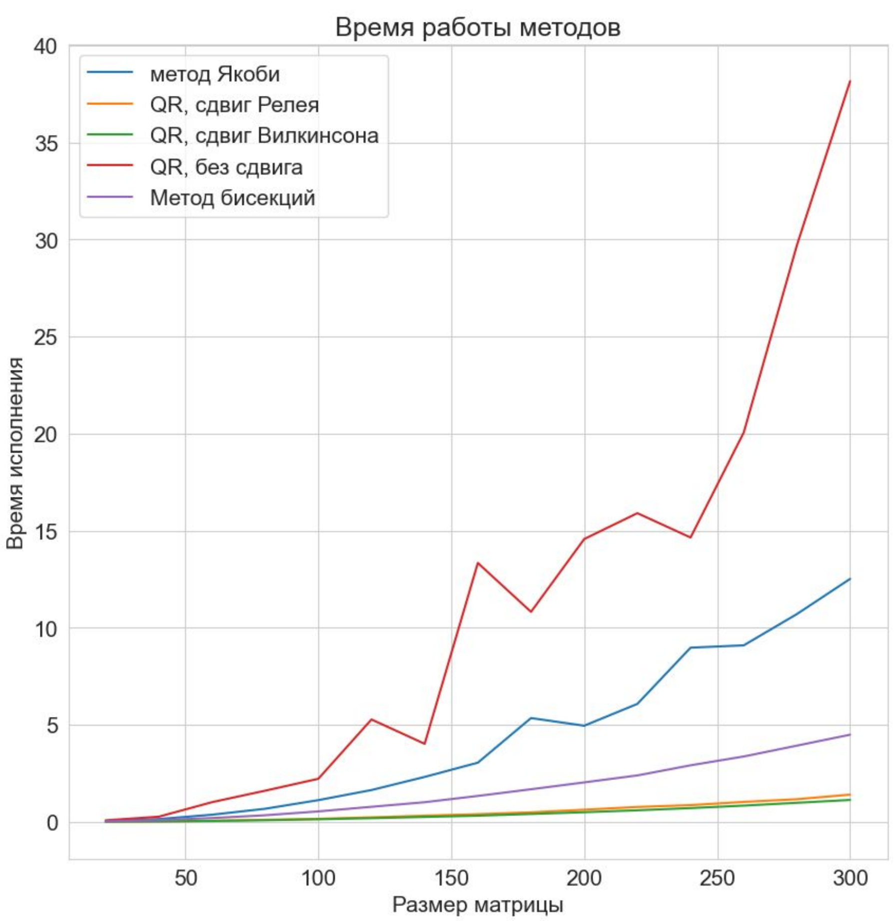
Можно выбрать максимальный недиагональный элемент, тогда сходимость будет наилучшей.

НО! Такой выбор стоит $O(n^2)$
Делаем цикл по всем элементам и кайфуем

Итого, **кубическая** сложность одного цикла. Его стоимость = половина времени **QR**-алгоритма с приведением к трехдиагональной матрице.

Нужно 5-15 циклов - очень медленно ;(





Сдвиги делают самый медленный QR-алгоритм самым быстрым!

Что еще можно было сделать

- Реализовать **неявные двойные сдвиги** для QR-алгоритма
- Изучить тему **мульти сдвигов**
- Реализовать метод “**Разделяй и властвуй**”
- Поработать с параллельной реализацией методов
- Изучить степенной сдвиг, и производные от него

Основные ссылки

- https://github.com/ShaeNaZar/AIM-numerical-linalg-project-1 - ссылка на github проекта
- https://twiki.cern.ch/twiki/pub/Main/AVFedotovHowToRootTDecompQRH/Golub_VanLoan.Matr_comp_3ed.pdf - Matrix computation, Charles Van Loan
- Вычислительная линейная алгебра, В.М. Вержбицкий
- Вычислительная линейная алгебра. Теория и приложения, Дж. Деммель