Вычисление псевдообратных матриц $A^\dagger = \lim_{\alpha \to 0}(\alpha I + A^* A)^{-1} A^*$

Команда: Линейная комбинация (далее ЛК..)

Состав: α_1(Олег) + α_2(Паша) + α_3(Даниил) + α_4(Артем)

Постановка задачи

- **о чем:** оцениваем методы вычислений псевдообратных матриц на практике (python + numpy)
- **зачем:** чтобы выявить наиболее устойчивые и/или быстрые методы для любых матриц и определенной структуры (например Эрмитовых)
- **качество по времени:** среднее затраченное время на вычисление
- **качество по устойчивости:** $\frac{\|\hat{x}-x\|}{\|x\|} \leq \frac{\mathrm{cond}(A)}{1-\mathrm{cond}(A)\frac{\|\Delta A\|}{\|A\|}}\left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta f\|}{\|f\|}\right)$
- **применяется в:** методе наименьших квадратов, задачах линейного программирования, стационарной вероятности в марковских цепях, нейронных сетях, обработке сигналов
- **примечание:** работаем исключительно с float64 (другого numpy не поддерживает в алгоритмах), над комплексными и вещественными числами

Описание методов

Для полноранговых матриц по столбцу

1. использование PLU разложения для вычисления $C = (A^* A)^{-1}$, $A^\dagger = CA^*$
2. использование QR разложения для A, с целью упрощения выражения $A^\dagger = R^{-1}Q^*$
3. использование SVD разложения для A, с целью упрощения выражения $A^\dagger = V\Sigma^{-1}U^*$

Для не полноранговых

Основная идея: использование разложений матриц в специальную структуру, чтобы можно было легко вычислить предел без необходимости интерполяции.

Такими примерами могут быть:

1. опять же SVD (**заместо деления на б.м. величину, оставляем нули в $\Sigma^\dagger$**, поскольку хотим min по норме решение)
2. разложение по собственным значениям (матрица **должна быть диаганализуема**)

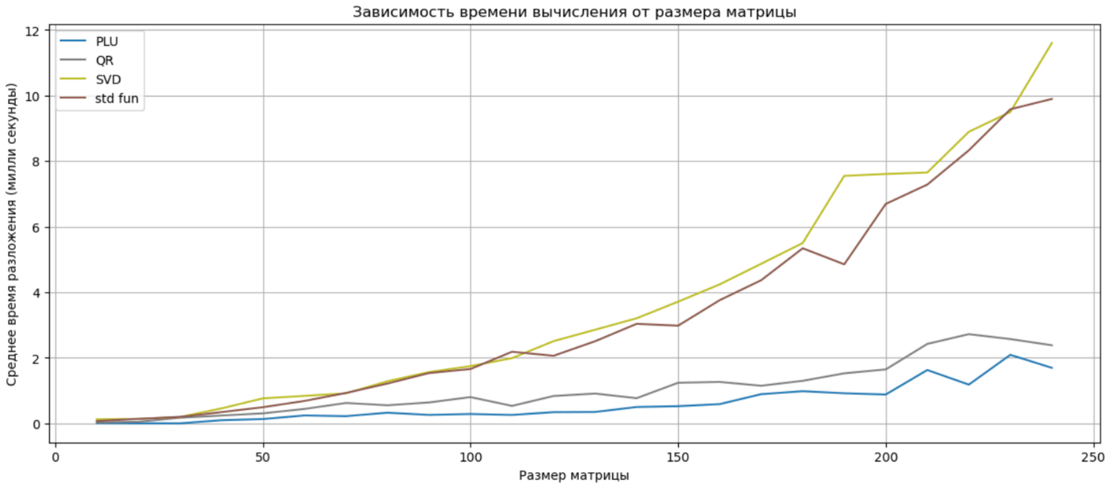
Более продвинутые методы будут разобраны **отдельно**

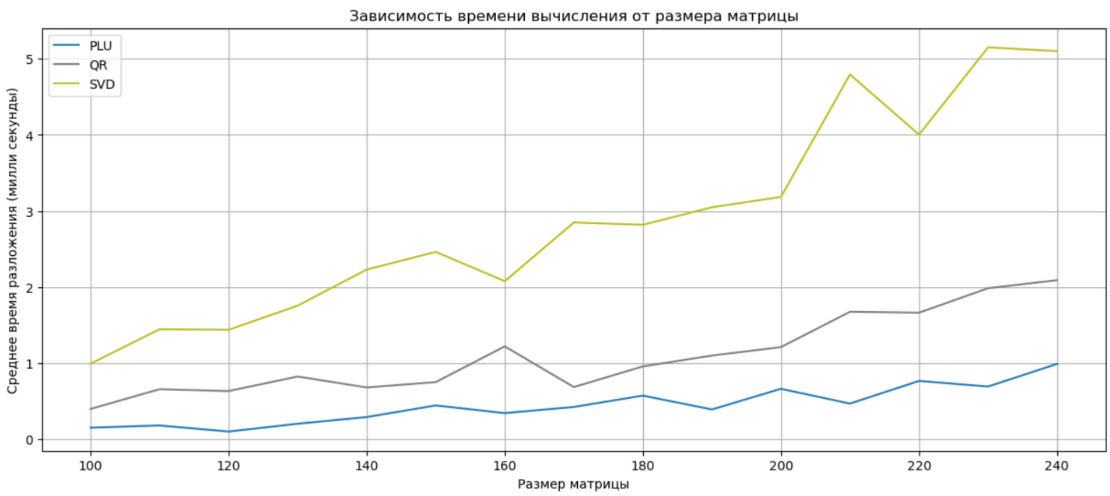
Некоторые свойства

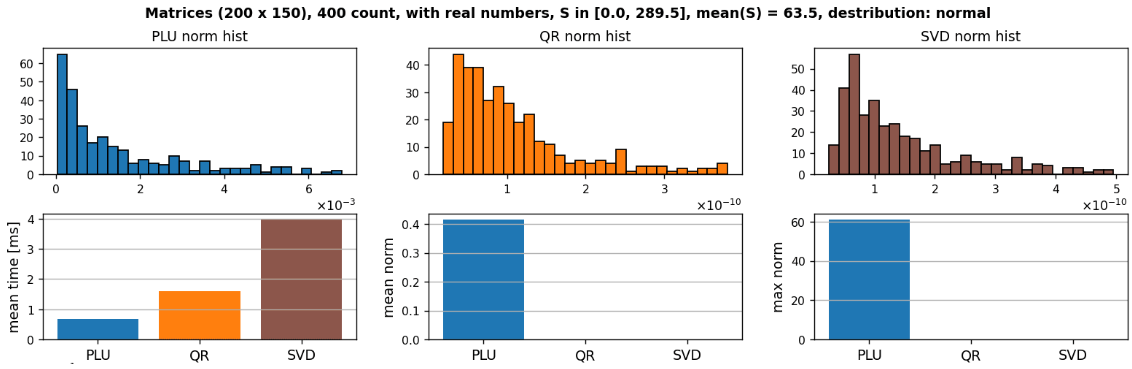
1. $(\alpha A)^\dagger = \alpha^{-1}A^\dagger$ - выносим скаляр

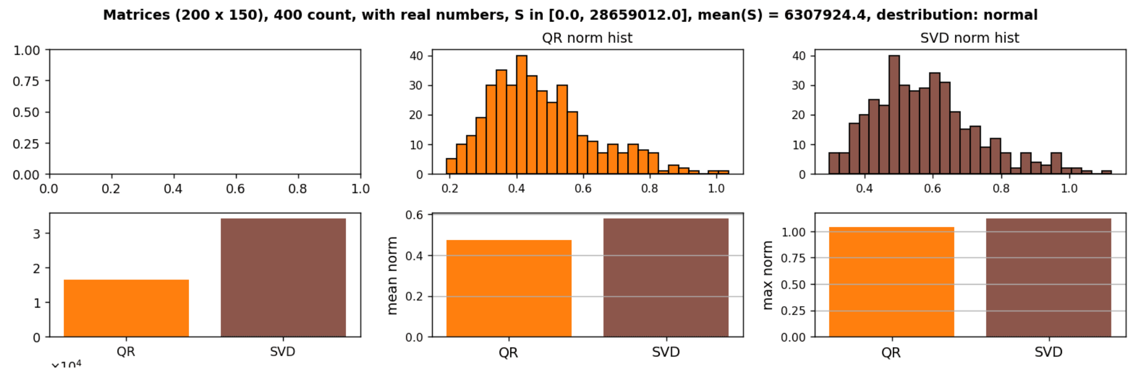
2. $(UA)^\dagger = A^\dagger U^*$ и $(AU)^\dagger = U^* A^\dagger$ - выносим унитарную матрицу (хорошо **сочетается с Эрмитовыми** матрицами, применяется с svd)
3. $A^\dagger = (A^* A)^{-1} A^*$ - полный столбцовый ранг
4. $A^\dagger = A^*(AA^*)^{-1}$ - полный **строчный ранг**
5. $(A^\dagger)^\dagger = A$ - дважны псевдообратная матрица равна исходной
6. $(A^\dagger)^* = (A^*)^\dagger$ - порядок $*$ и $\dagger$ не важен
7. $A^\dagger = A^*(AA^*)^\dagger$ - **вычисление через симметричную** (хороший вариант, если умеем быстро диаганализировать)
8. $A^\dagger A = AA^\dagger$, если A **- нормальная**

Результаты svd vs qr vs plu









Результаты для метода Гревиля

Пусть матрица A получается из A_1 дописыванием столбца α справа: $A = (A_1, \alpha)$

1. Случай $A_1 \cdot A_1^\dagger \cdot \alpha = \alpha$:

Введем: $\beta = \frac{(E - A_1 \cdot A_1^\dagger) \cdot \alpha}{\|(E - A_1 \cdot A_1^\dagger) \cdot \alpha\|_2^2}$

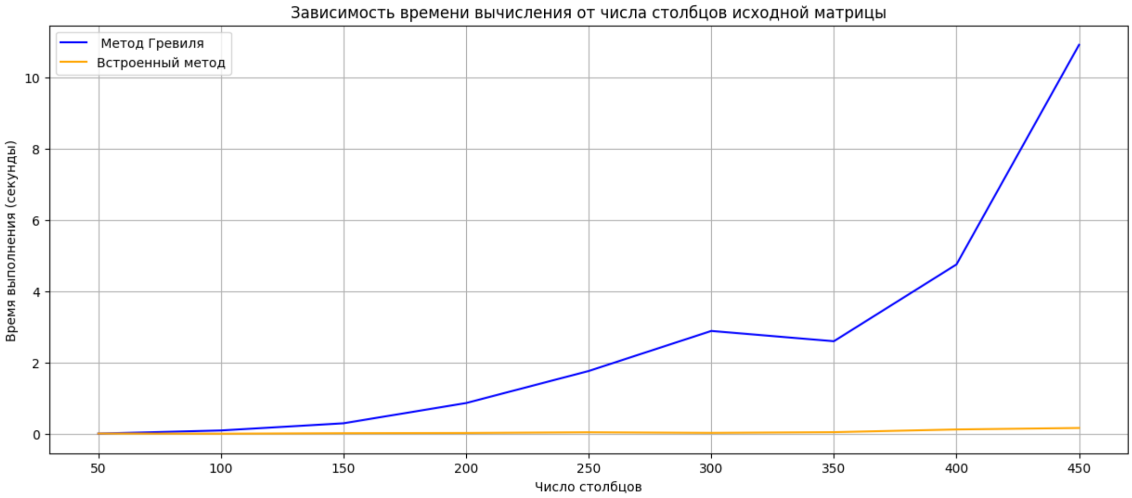
Тогда псевдообратная матрица: $A^\dagger = \begin{pmatrix} A_1^\dagger \cdot (E - \alpha \cdot \beta^T) \\ \beta^T \end{pmatrix}$

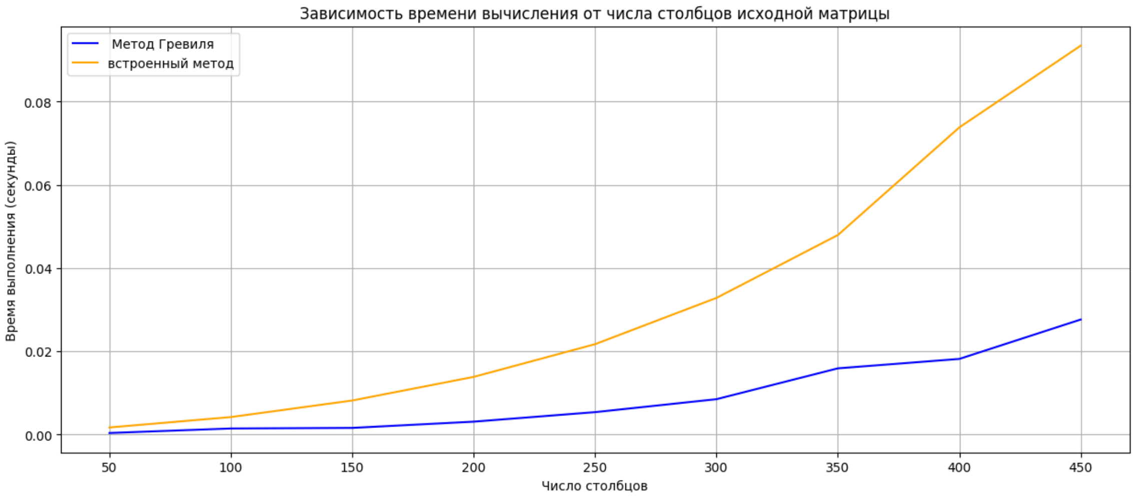
1. Случай $A_1 \cdot A_1^\dagger \cdot \alpha \neq \alpha$:

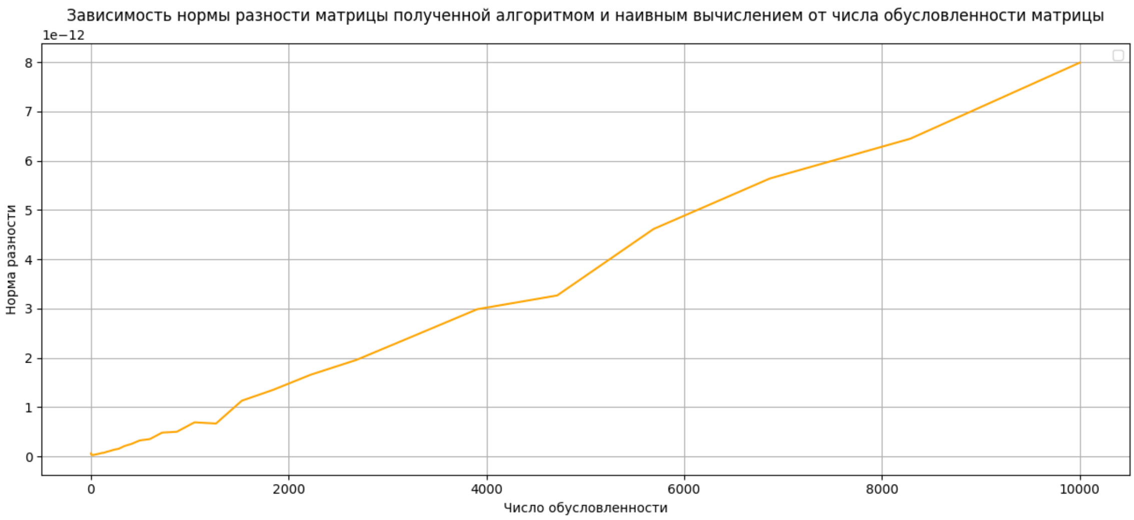
Введем: $\gamma = \frac{A_1^{\dagger, T} \cdot A_1^\dagger \cdot \alpha}{1 + \|A_1^\dagger \cdot \alpha\|_2^2}$

Тогда псевдообратная матрица: $A^{\dagger} = \begin{pmatrix} A_1^{\dagger} \cdot (E - \alpha \cdot \gamma^T) \\ \gamma^T \end{pmatrix}$

Таким образом мы можем взять первый столбец и последовательно добавляя новые столбцы будем производить вычисления псевдообратных матриц.

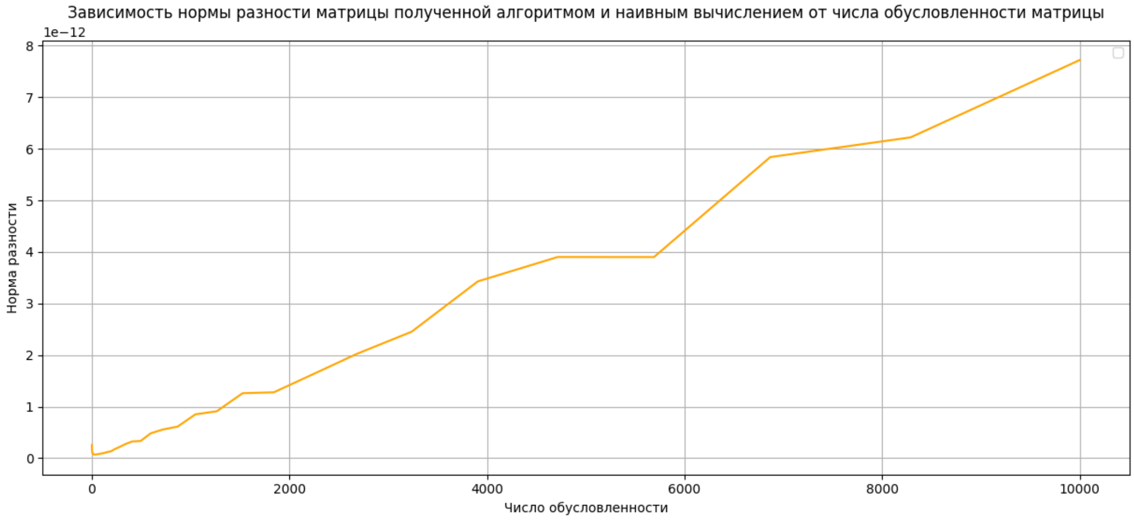


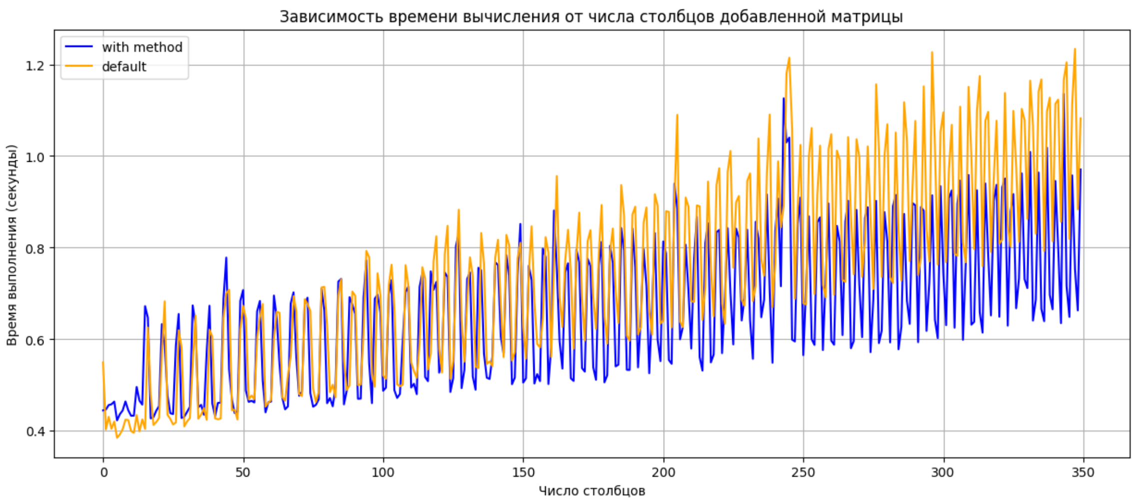




Результаты для блочных матриц

$$\begin{bmatrix} H_1 & H_2 \end{bmatrix}^+ = \begin{bmatrix} E & H_1^+ H_2 \\ H_2^+ H_1 & E \end{bmatrix}^{-1} \begin{bmatrix} H_1^+ \\ H_2^+ \end{bmatrix}^+$$





Выводы

По стандратным методам псевдообращения - ожидание совпало с реальностью.

Были найдены границы их применения и скорость работы на которую можно полагаться.

Наиболее универсальным выбором - выступает svd, наиболее выгодным для полноранговых матриц - показало себя QR разложение, а если вы уверены, что число обусловленности достаточно мало - ваш выбор PLU разложение.

Метод Гревиля является удобным, если известна псевдообратная матрица для исходной и хотим найти пседообратную к матрице, полученной из исходной добавлением строки или столбца.

Для блочных матриц удобен алгоритм, который является обобщением одного шага метода Гревиля. Он обладает вычислительными преимуществами по сравнению с аналогами в случае, если известна псевдообратная матрица для исходной, а также в некоторых случаях, когда псевдообратная матрица для исходной неизвестна, - для матриц больших размеров, у которых сумма числа столбцов соотносится с их числом строк.

Ссылка на гитхаб

https://github.com/Acool4ik/NLA-calculation-of-pseudoinverse-matrices (https://github.com/Acool4ik/NLA-calculation-of-pseudoinverse-matrices)