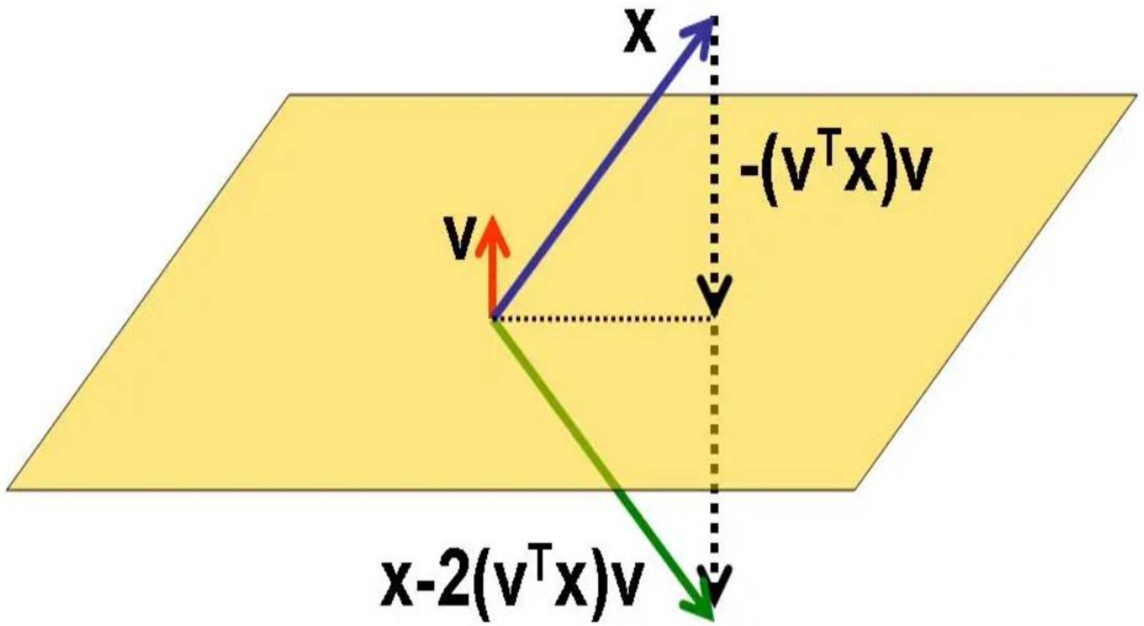
A block Cholesky-LU-based QR factorization for rectangular matrices
Team 007

Link to GitHub

In this presentation, we will explore the powerful technique of QR factorization using Householder method recursively and cutting off the recursion at a small enough block and solving it using the Cholesky-LU method specifically designed for rectangular full rank matrices.



Skoltech
Skolkovo Institute of Science and Technology

Motivation

1

Numerical Stability

Since the method uses Orthogonal projections , it gets some of the stability as norms are preserved in some of the operations

2

Computational Efficiency

Block Based computation are more efficient as the maximimise cache hit rate in data retrieval

3

Best of both Worlds

This method combines the efficient part of the stability of using HouseHolder methods with the efficiency in computing The bBock Cholesky LU factorisation

Cholesky-LU-based QR Factorization for Rectangular Matrices

1

Block Cholesky Decomposition

We introduce a modified algorithm that incorporates the Cholesky decomposition into the QR factorization for rectangular matrices.

2

LU Factorization

By employing the LU factorization, we enhance the efficiency of the QR factorization process.

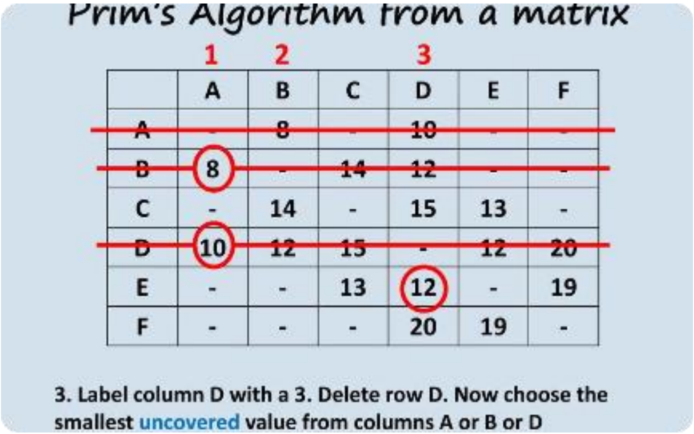
3

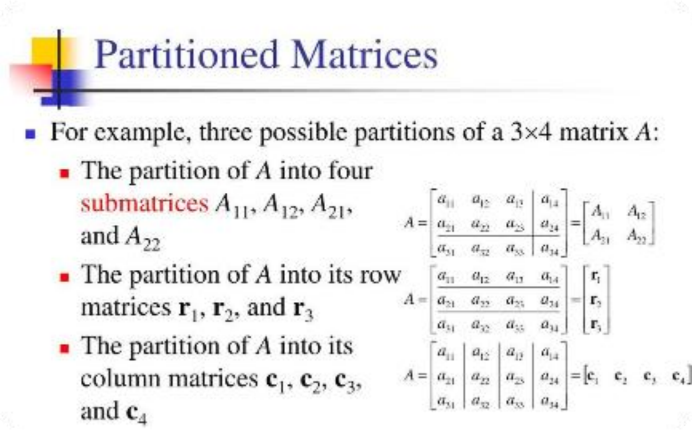
Improved Result Accuracy

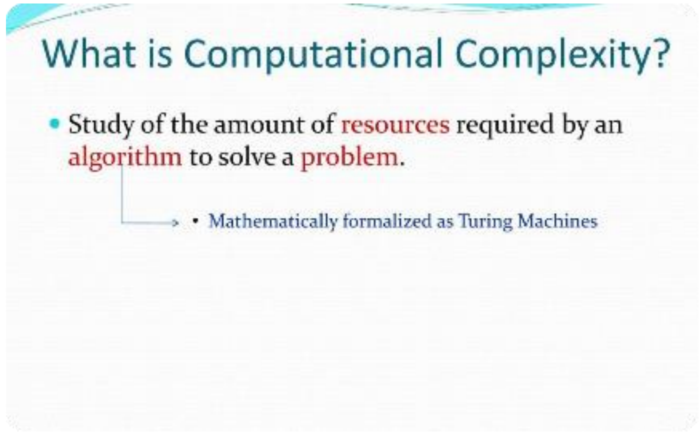
Our approach minimizes errors and maintains high accuracy in the presence of skinny rectangular matrices.

Skoltech
Skolkovo Institute of Science and Technology

Block-based Algorithms for QR Factorization







Block Householder Transformations

We present a new algorithm that uses block-based Householder transformations to improve the efficiency of QR factorization.

Matrix Partitioning

The method incorporates block-wise matrix partitioning to optimize the factorization process for larger, rectangular matrices.

Computational Complexity

We visualize the empirical time taken in the block-based algorithms and demonstrate their advantages compared to traditional methods.

Skoltech
Skolkovo Institute of Science and Technology

Problem Statement

Algorithm 1

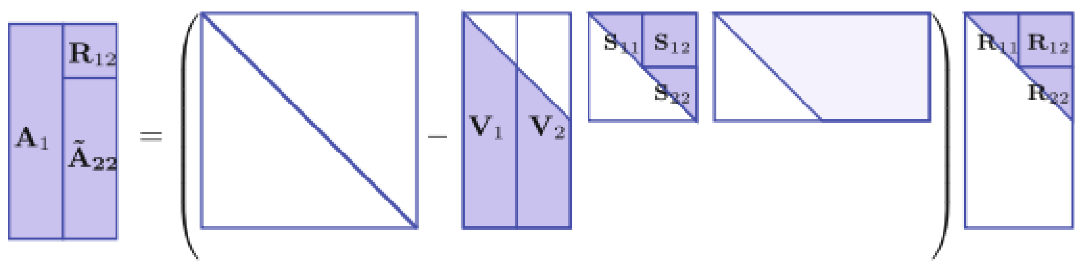
We consider $m \geq n$

Recursive Householder-based QR factorization of $A \in \mathbb{R}^{m \times n}$ *with orthogonal* factor of the form $Q = I - VSV^T$.

Input $A \in \mathbb{R}^{m \times n}$

Output $V, R \in \mathbb{R}^{m \times n}$ $S \in \mathbb{R}^{n \times n}$ representing the QR factor of A

$$\begin{pmatrix} A_1 & A_2 \end{pmatrix} = \left(I - \begin{pmatrix} V_1 & V_2 \end{pmatrix} \begin{pmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{pmatrix} \begin{pmatrix} V_1 & V_2 \end{pmatrix}^T \right) \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \\ 0 & 0 \end{pmatrix},$$



Skoltech
Skolkovo Institute of Science and Technology

Algorithm 2

We consider $m \geq n$
Block Cholesky-LU-based QR factorization of $\mathrm{A} = (A_{\square}^{T}, A_{r}^{T})^{T}$ *with orthogonal* factor of the form $Q = I - VSV^{T}$.

Input $A \in \mathbb{R}^{m \times n}$
Output $V, R \in \mathbb{R}^{m \times n}$ $S, R \in \mathbb{R}^{n \times n}$ representing the QR factor of A.
The block form can be inferred from the equation below

$$\begin{pmatrix} A \\ A_r \end{pmatrix} = A = QR = (Y|Z)\begin{pmatrix} R \\ 0 \end{pmatrix}$$

Algorithm 3 (Proposed Method)

We consider $m \geq n$
Recursive block QR factorization of $A \in \mathbb{R}^{m \times n}$ with orthogonal factor of the form $Q = I - VSV^T$ with user specified input $1 < n < k$

Input $A \in \mathbb{R}^{m \times n}$
Output $V, R \in \mathbb{R}^{m \times n}$ $S \in \mathbb{R}^{n \times n}$ representing the QR factor of A.

Psuedocode

If $1 < n < k$
    Perform Algorithm 2
else
    Perform algorithm 1
endif

Skoltech
Skolkovo Institute of Science and Technology

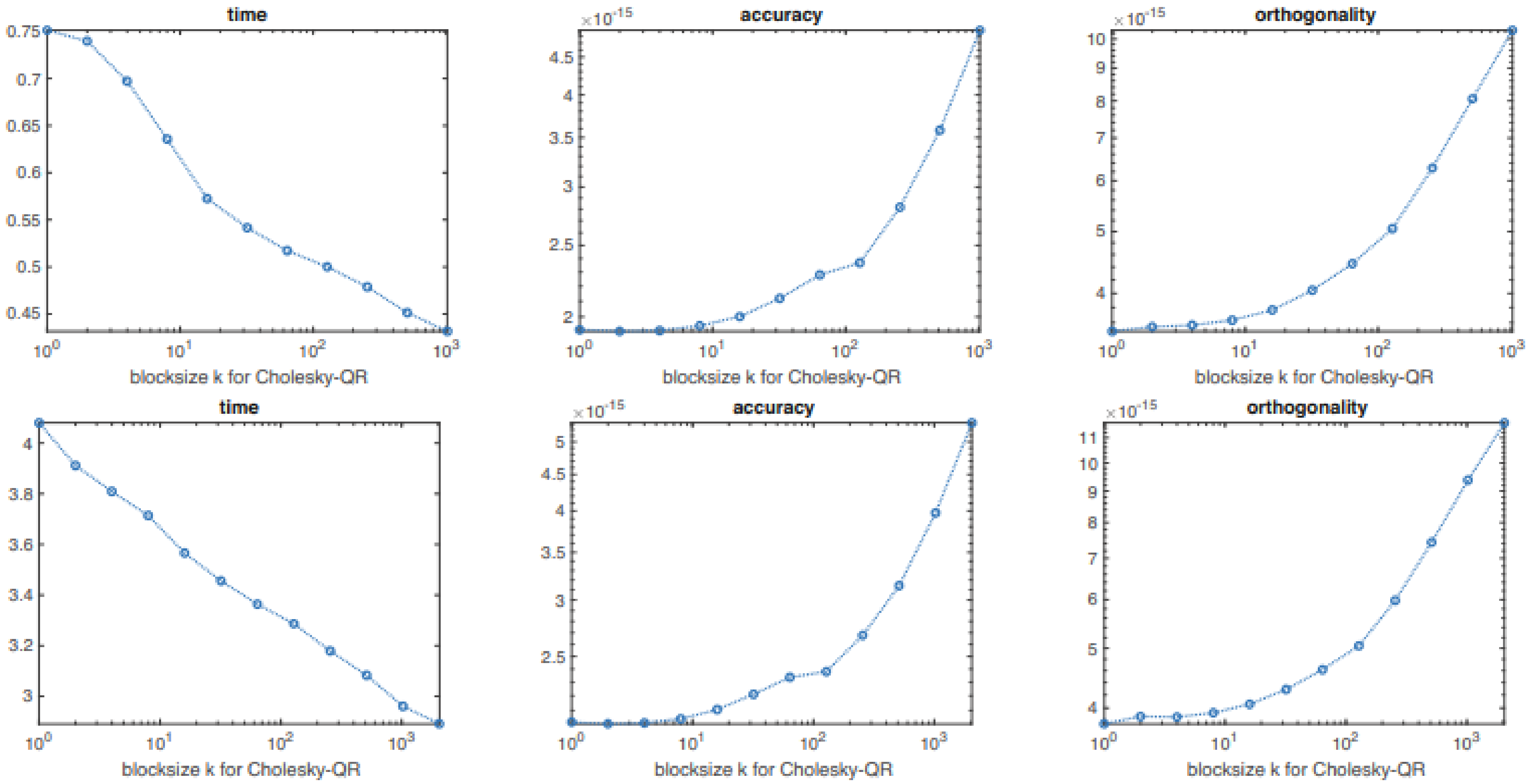
Problem Statement

An algorithm that factors tall rectangular matrices into QR orthogonal factors that is computationally efficient.

The known algorithms are either very fast and unstable(eg Cholesky Block LU(Algorithm 2)) or quite slow and stable(eg continuing the HouseHolder method down to a single column (Algorithm 1))

So our task is to find an algorithm that takes advantage of strength of both algorithms

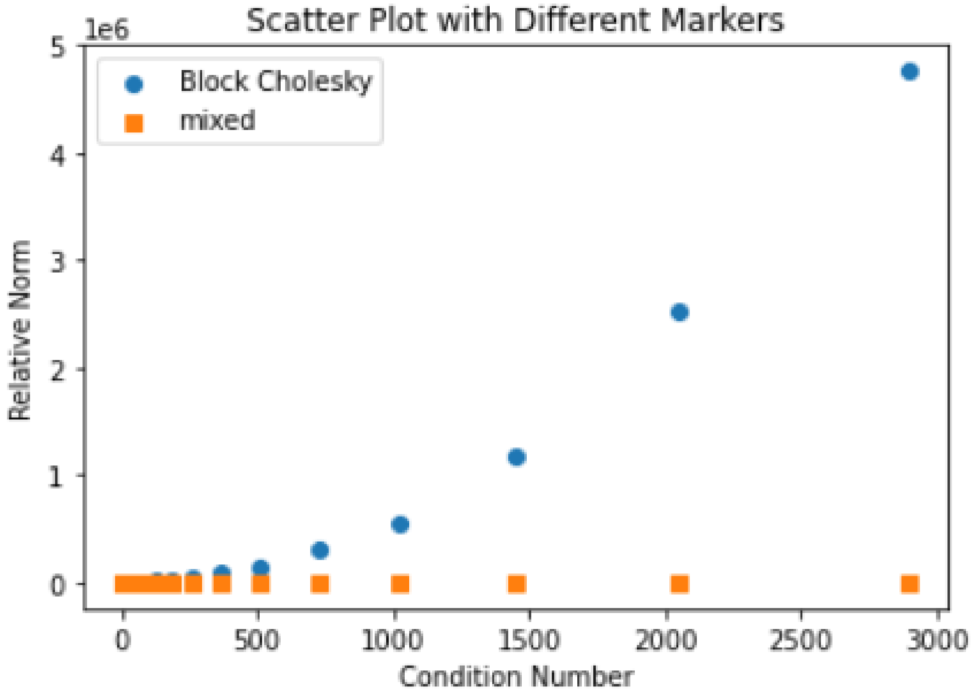
Numerical Results

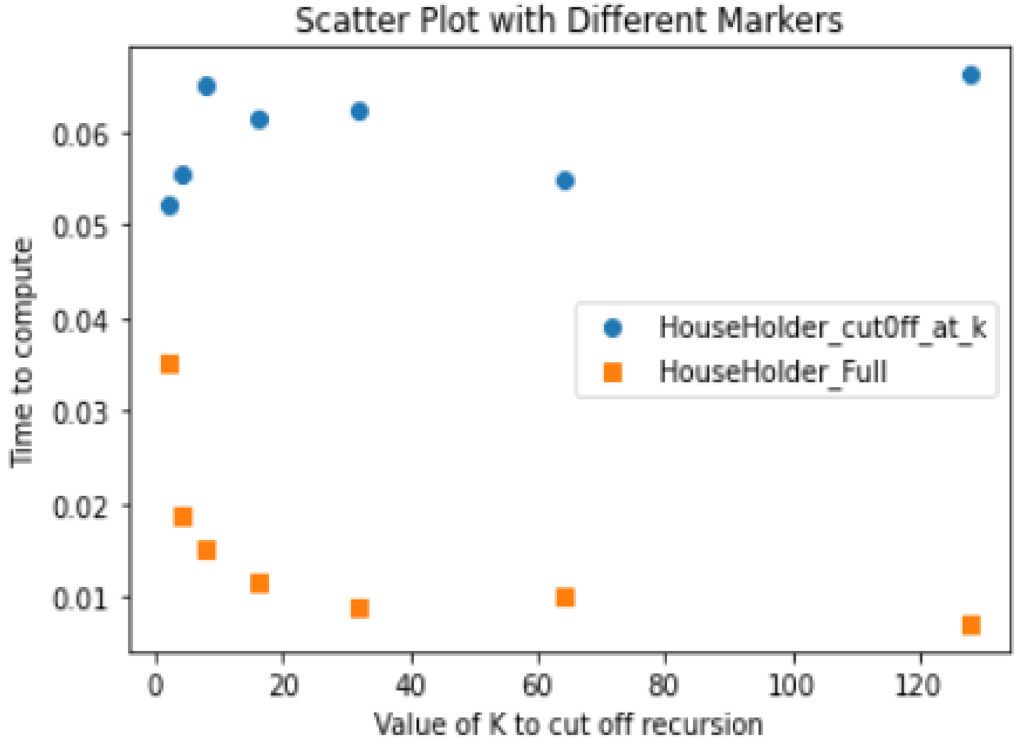


Skoltech
Skolkovo Institute of Science and Technology

Results for a dense 2500 × 2500 matrix (top row) and a dense 5000 × 5000 matrix (bottom row), both filled with random entries

Numerical Results





Comparison of Algorithm 3 with Algorithm 2 and 1 on stability and computational time respectively for m = 2000 and n = 500

Skoltech
Skolkovo Institute of Science and Technology

Team Contribution

- Holdings Ogon - Implemented algorithm 1 and helped with writing the condition number function
- Joshua Udobang - Did the plot graphing and observed the empirical results
- Nwachukwu Mmesomachi - Implemented Algorithm 3 and verified the mathematical correctness. Wrote the README file
- Okechukwu Okeke - Implemented algorithm 2 and modularized the code
- All contributed to making the presentation

References

1. Le Borne S. A block Cholesky-LU-based QR factorization for rectangular matrices. Numer Linear Algebra Appl. 2023;30(5):e2497
2. Barlow J. Block modified Gram-Schmidt algorithms and their analysis. SIAM J Matrix Anal Appl. 2019;40:1257–90.
3. Carson E, Lund K, Rozloznik M, Thomas S. Block Gram-Schmidt algorithms and their stability properties. Linear Algebra Appl. 2022;638:150–95.
4. Terao T, Ozaki K, Ogita T. LU-Cholesky QR algorithms for thin QR decomposition. Parallel Comput. 2020;92:102571. https://doi.org/10. 1016/j.parco.2019.102571 4.
5. Golub G, Van Loan C. Matrix computations. 3rd. ed. London: John Hopkins; 1996. 5.
6. Kressner D, Susnjara A. Fast QR decomposition of HODLR matrices, 2018. https://arxiv.org/abs/1809.10585 6.
7. Le Borne S. Block computation and representation of a sparse nullspace basis of a rectangular matrix. Linear Algebra Appl. 2008;428:2455–67

Skoltech
Skolkovo Institute of Science and Technology