# Task-based implementation of QR factorization and SVD

Author: Ing. Vít Břichňáč | Supervisor: Ing. Jakub Šístek, Ph.D.

FACULTY OF INFORMATION TECHNOLOGY CTU IN PRAGUE

## Motivation

The **QR factorization** and **Singular Value Decomposition (SVD)** are matrix factorizations with applications in data analysis, information retrieval, numerical simulations and many other fields.

In many of those applications (such as the PCA method), the factorized matrices tend to have significantly more rows than columns, called **tall-and-skinny** matrices.

Commonly used libraries for numerical linear algebra usually provide implementations of factorization routines that are heavily optimized for generic matrices, while not always employing algorithms that best utilize the special properties of tall-and-skinny matrices.

## Mathematical background

The QR factorization and the SVD of a matrix $A \in \mathbb{R}^{m,n}$ are given by the equations

$$A = QR \quad \text{and} \quad A = U\Sigma V^T$$

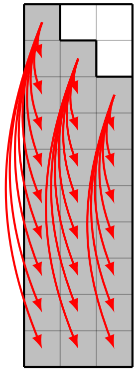
respectively, where:

- $Q \in \mathbb{R}^{m,n}$, $U \in \mathbb{R}^{m,\min(m,n)}$ and $V \in \mathbb{R}^{n,\min(m,n)}$ are matrices with orthonormal columns,
- $R \in \mathbb{R}^{n,n}$ is an upper triangular matrix, and
- $\Sigma \in \mathbb{R}^{\min(m,n),\min(m,n)}$ is a diagonal matrix with nonnegative diagonal elements in a decreasing order.

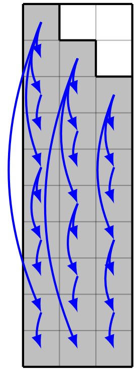
## Elimination schemes

To achieve high performance on modern multicore processors with vector registers, state-of-the-art algorithms for matrix factorization typically split the factorized matrix into blocks. Operations are then performed on those blocks separately, allowing for dynamic parallelization.

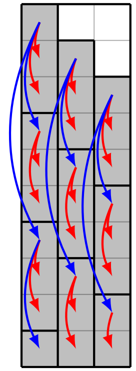
The performance of the algorithms can be heavily influenced by the order in which the blocks are processed (the so-called **elimination scheme**). While some elimination schemes work better for certain matrix types, it is optimal for the elimination scheme to adapt to the matrix shape.



The *flat tree* elimination scheme, most suitable for square matrices.



The *binary tree* scheme, suitable for tall-and-skinny matrices.



The *superblock binary tree* scheme adapts to the matrix shape.

## Results

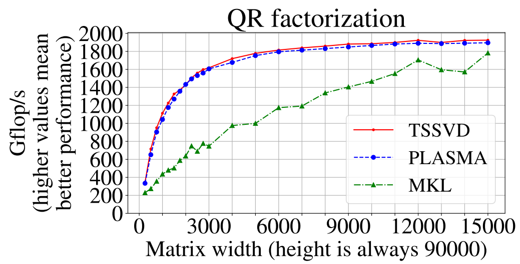
The performance of *TSSVD* was compared with the vendor-provided *Intel OneApi MKL* and the *Arm Performance Libraries*, as well as the state-of-the-art *PLASMA* library for numerical linear algebra operations on multicore shared-memory systems.

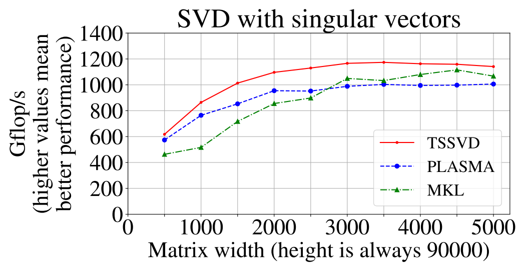
| Operation | TSSVD performance |
|---|---|
| QR factorization | comparable to PLASMA, outperforms others |
| SVD w/o singular vectors | outperforms all three for most tested matrix sizes |
| SVD with singular vectors | outperforms all three for all tested matrix sizes |

## Implementation

The implementation was released as an open-source library TSSVD under the 3-clause BSD license. The library was written in the C language using the OpenMP task system for parallelization. Its correctness was checked using an extensive test set.

## Performance plots (on Intel CPU)




## References

[1] DONGARRA, Jack; GATES, Mark et al. The Singular Value Decomposition: Anatomy of Optimizing an Algorithm for Extreme Scale. *SIAM Review*. 2018, vol. 60, no. 4, pp. 808–865.

[2] BŘICHŇÁČ, Vít; ŠÍSTEK, Jakub. Performance of parallel QR factorization methods on the NVIDIA Grace CPU Superchip. In: *Programs and Algorithms of Numerical Mathematics 22: Proceedings of Seminar*. Institute of Mathematics of the Czech Academy of Sciences, Prague, 2025, pp. 29–40.