What is SVD (Singular Value Decomposition):

Given a matrix A of size $m \times n$, the SVD decomposes A into the product of three matrices:

$$A = U\Sigma V^T$$

Where:

U : An $m \times m$ orthogonal matrix (columns are orthonormal eigenvectors of AA^T)

Σ : An $m \times n$ diagonal matrix containing the singular values (non-negative values sorted in decreasing order)

V^T : An $n \times n$ orthogonal matrix (rows are orthonormal eigenvectors of AA^T)

Why it is important:

SVD is essential for dimensionality reduction, noise filtering, and matrix approximations, widely applied in data science and machine learning. Parallel SVD is crucial for handling large-scale datasets efficiently, enabling faster computation and scalability in high-dimensional applications.

Method:

Our project focuses on implementing a parallelized Singular Value Decomposition (SVD) using the ***one-sided Jacobi method***.

Core of Jacobi method:

The method transforms the symmetric matrix A into a diagonal matrix A′, whose diagonal entries are the eigenvalues of AA^T, using a sequence of rotations (shown in Figure 1).

$$G(i, j, \theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

Figure 1

The rotation matrix $G(i, j, \theta)$ is a rotational matrix that operates only on the $i - th$ and j$-th$ rows/columns to make them orthogonal to each other. Sequentially, we can make all the columns orthogonal to each other. In this way, we would be able to retrieve $U * \Sigma$, and our final rotational matrix would be V.

Parallelization Strategy:

The independence of Jacobi rotations makes the method highly amenable to parallel computation. Each rotation involves only two columns, and different column pairs can be processed simultaneously.

OpenMP:
MPI:
CUDA:
Hybrid Parallelization:
By benchmarking each parallelization approach and their hybrid implementation, the project provides insights into the efficiency of these methods for large-scale SVD computations.