**Parallelization of Singular Value Decomposition**

Meivenkatkumar Lakshminarayanan, Nayanthara K. Jayadev

* Singular Value Decomposition
  + - Powerful matrix factorization technique that decomposes a matrix into three other matrices, revealing important structural aspects of the original matrix.
    - A wide range of applications, including signal processing, image compression, and dimensionality reduction in machine learning, and **quantum chemistry.**

A diagram of a square with letters and numbers

Description automatically generated

* Quantum Chemistry applications
  + - Wave functions provide us probability of finding an electron at a certain position in space.
    - In one-electron systems, orbitals are wavefunctions.
    - The wave functions of many-electron systems depend on the  
      coordinates of all electrons.
    - Concept of orbital gets murkier.
    - Introduction of orbitals useful for different processes (example: Dyson orbitals, Natural transition orbitals)
* Auger decay
  + - One valence electron fills the core-hole, and another gets ejected into  
      the continuum.

A diagram of a process

Description automatically generated

* + - The decay width can be calculated as,

A close-up of a text

Description automatically generated

* + - Two-body Dyson amplitudes contains information about Auger decay from the bound domain.

A black and white text

Description automatically generated

* + - **The most compact representation of such third-rank tensors can be obtained using a two-step singular value decomposition giving natural Auger orbitals.**

A diagram of a square with a red line and blue line

Description automatically generated

* + - However, the SVD of bigger matrices is very slow compared to other steps in the algorithm proposed by Krylov and coworkers.1

* Aim
  + - Parallelization of singular value decomposition of third rank tensors relevant to Auger decay.
    - Comparison between different available algorithms and measure the speed up compared to a serial run.
    - Parallelize the SVD code using MPI, OpenMP, and CUDA, and compare it with the existing algorithms for two-body Dyson tensors.
* Preliminary investigation: Existing Algorithms
  + - Using PyParSVD: Python Parallel Singular Value Decomposition.

PyParSVD is based on algorithms using distribution or partitioned SVD as shown by Wang *et al.2 .* The process can be distributed into much smaller tasks over multiple processors in parallel, computational time can be drastically reduced. In this paper, we put forth a novel partitioned method for generating the SVD basis from data. This method preserves the distributed nature of the data and takes advantage of parallelism for computation. Additionally, it greatly reduces subtask communication volume.

<https://github.com/Romit-Maulik/PyParSVD#Wang-et-al-2016>

* + - Parallel SVD using Jacobi’s rotations, implemented in OpenMP.

<https://github.com/lixueclaire/Parallel-SVD/blob/master/OMP_SVD.cpp>

* + - A parallelized implementation of Principal Component Analysis (PCA) using Singular Value Decomposition (SVD) in OpenMP for C. The procedure used is Modified Gram Schmidt algorithm.

<https://github.com/arneish/parallel-PCA-openmp>