Eigenvalue problem

Given a matrix $A$ of size $n \times n$, the eigenvalue problem consists of finding eigenvalues $\lambda_i \in \mathbb{C}$ and eigenvectors $x_i \in \mathbb{C}^n$ such that $Ax_i = \lambda_i x_i$.

Suppose from now on that $A \in \mathbb{R}^n$ and $A \neq A^T$.

Algorithms

The overall goal is to apply a series of similarity transformations to $A$.

**Hessenberg reduction**: $A$ is first reduced to upper Hessenberg form $A = QHQ^T$.

This is done in order to greatly accelerate the subsequent QR algorithm.

**QR algorithm**: $H$ is then reduced to Schur form $H = QSQ^T$ using multishifts and aggressive early deflation (AED).

The eigenvalues $\lambda_i \in \mathbb{C}$ can be determined from the diagonal $1 \times 1$ and $2 \times 2$ blocks of $S$.

**Eigenvalue reordering**: In some cases, we want to reorder the Schur form $S$ such that a selected cluster of eigenvalues appears in the leading diagonal blocks of the updated Schur form $S = QSQ^T$.

**Eigenvectors**: The eigenvectors $x_i = Q^Ty_i$ can be solved from $(S - \lambda I)y_i = 0$.

See Carl Christian’s poster.

Task-based approach

The overall idea is to describe each algorithm in terms of a task graph.

The runtime system can schedule some tasks to GPUs.

The runtime system can also handle MPI communications.

Shared memory (SM) performance

Preliminary performance of the Schur reduction (QR) component compared to LAPACK + parallel BLAS and ScalAPACK.

Distributed memory (DM) performance

Scalability of the eigenvalue reordering component (35% of the diagonal blocks selected).

GPU performance

Performance of the Hessenberg reduction component compared to LAPACK, MAGMA GPU.

Conclusions

- The library aims to provide the complete chain of algorithms for standard and generalized non-symmetric eigenvalue problems.
- The same software is designed to run on both shared memory and distributed memory machines.
- The library also aims to accelerate the performance using GPUs.

References


Acknowledgements

All computational experiments were performed using resources available at High Performance Computing Center North (HPC2N).