SI2-SSE Enhancing the PRIMME Software with new Methods and Functionality for Eigenvalue and SVD problems

Andreas Stathopoulos, College of William and Mary, Williamsburg, Virginia, USA

Introduction

- PRIMME: Preconditioned Iterative MultiMethod Eigensolver
  - Both exterior and interior Hermitian eigenproblems
  - Over 12 methods accessible through PRIMME
  - Near optimal methods GD+k and JDDMR
  - Allows for preconditioning
  - Full set of defaults and auto-tuning for end-users
  - Full customizability for expert users
  - Parallel, high performance implementation
  - C and Fortran interfaces

- Impact
  - Widely used in a variety of applications:
    - Electronic structure and materials science
    - Structural engineering
    - Lattice QCD
    - Spectral graph partitioning
  - Used by several groups:
    - National labs, academic research, industry

State-of-The-Art Eigensolvers

- Current state-of-the-art solvers and methods

- Performance

- Improvements in PRIMME version 1.2 (Feb '15)
  - A Fortran compiler is no longer required for building PRIMME
  - Fixed some uncommon issues with the F77 interface
  - PRIMME can be called now multiple times from the same program
  - Performance improvements in the QMR inner solver, especially for complex
  - Fixed a couple of bugs and tuned the locking functionality
  - Unique random seeds per parallel process for up to 4096^3 processes
  - For the DYNAMIC method, fixed issues with initialization and synchronization decisions across multiple processes
  - Other performance and documentation improvements
  - MATLAB interface through MEX ready to be finalized

Proposed Software Development

- Our goal: extend PRIMME’s efficiency and robustness to highly interior eigenvalues, generalized eigenvalue and SVD problems

User Interface

- MATLAB, Hype, Trilinos optional interfaces

PRIMME architecture. Red boxes are the proposed components.

User provided functions

- Eval targeting
- Convergence test

Methods

- GD main iteration implementing parameterized methods
- GD+k, JDDMR, LOPBCG, RQE, etc.

Building Blocks

- Locking
- Restarting
- Convergence test
- Orthogonalization
- RayleighRitz
- Adaptive QR

Libs

- Interface to BLAS-LAPACK
- Link to numerical setup of MATLAB, Hype, Trilinos

Evolve targeting

- Convergence test

- precond = MFS, MPS

Multi-method agent choosing between eigenvalue and native cvd methods

Building Blocks

- Locking
- Restarting
- Convergence test
- Orthogonalization
- RayleighRitz
- Adaptive QR

Evaluation targeting

- Convergence test

- precond = MFS, MPS

Multi-method agent choosing between eigenvalue and native cvd methods

Comparison of Different Approaches

- Our approach: a hybrid, two-stage SVD method

- GD+k on C
  - Fast for largest SVs
  - Slow for smallest SVs
  - Achieve accuracy of \( O(\epsilon) \)

- JDDMR on B
  - Slower for largest SVs
  - Extremely slow for smallest SVs
  - Achieve accuracy of \( O(\epsilon) \)

- PHSVD on A
  - Fast for largest SVs
  - Similar to C but exhibits irregular convergence for smallest SVs
  - Achieve accuracy of \( O(\epsilon) \)

Experimental results:

- Matvec and time ratio when seeking 1 smallest on square matrices
- Matvec and time ratio when seeking 10 smallest on rectangular matrices

State-of-the-art Eigensolvers

<table>
<thead>
<tr>
<th>Software</th>
<th>Methods</th>
<th>Lang</th>
<th>MPI</th>
<th>Interior</th>
<th>GHEP</th>
<th>SVD</th>
<th>Refs</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIMME</td>
<td>multiparameters</td>
<td>C</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>66</td>
</tr>
<tr>
<td>SLEPc</td>
<td>multiparameters</td>
<td>C</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>254</td>
</tr>
<tr>
<td>ANASAzi k, GD, LOPBCG, IRTR</td>
<td>C++</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>BLOPEX</td>
<td>LOPBCG</td>
<td>C</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>35</td>
</tr>
<tr>
<td>JADAMILUL</td>
<td>JDCG with MLU</td>
<td>F77</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>44</td>
</tr>
<tr>
<td>FEAST</td>
<td>FEAST</td>
<td>C</td>
<td>SMP</td>
<td>Y</td>
<td>N</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>PROPACK</td>
<td>Lanczos</td>
<td>F77</td>
<td>N</td>
<td>N</td>
<td>N/A</td>
<td>Y</td>
<td>195</td>
</tr>
<tr>
<td>SVDPACK</td>
<td>Lanczos</td>
<td>F77</td>
<td>some</td>
<td>N</td>
<td>N/A</td>
<td>Y</td>
<td>221</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Matrix</th>
<th>Methods</th>
<th>ANDrews</th>
<th>cdff1</th>
<th>or56R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>MV</td>
<td>Sec</td>
<td>MV</td>
<td>Sec</td>
</tr>
<tr>
<td>PRIMME JDDMR</td>
<td>768</td>
<td>4.1</td>
<td>4650</td>
<td>35</td>
</tr>
<tr>
<td>PRIMME GD+k</td>
<td>568</td>
<td>7.8</td>
<td>4383</td>
<td>81</td>
</tr>
<tr>
<td>Anasazi IRTR</td>
<td>2620</td>
<td>25.3</td>
<td>29090</td>
<td>320</td>
</tr>
<tr>
<td>Anasazi LOPBCG</td>
<td>1360</td>
<td>22.6</td>
<td>10610</td>
<td>209</td>
</tr>
<tr>
<td>Anasazi LOPBCG1</td>
<td>2291</td>
<td>28.2</td>
<td>17724</td>
<td>282</td>
</tr>
<tr>
<td>Anasazi GD1</td>
<td>2742</td>
<td>34.6</td>
<td>65762</td>
<td>1033</td>
</tr>
</tbody>
</table>