The partial Hermitian eigenvalue and singular value problems for large, sparse matrices

Andreas Stathopoulos

Computer Science Department
College of William and Mary

Acknowledgment: NSF CCF, NSF SI2, DOE SciDAC
The eigenvalue and singular value problems

Given $A$ Hermitian, find $nev$ eigenpairs:

$$Ax_i = \lambda_i x_i, \ i = 1 \ldots nev$$

Given $A$ any square or rectangular matrix, find $nsv$ singular triplets:

$$Av_i = \sigma_i u_i, \ i = 1 \ldots nsv$$

- One of the dimensions of $A$ can be $O(10^6 - 10^8)$
- $A$ is sparse or provided through a matvec function
Why this problem?

- Quantum mechanics (Schrödinger equation) a successful model of the world!

- Also macroscopic phenomena that involve vibrations/frequencies
  
  Structural engineering
  Fluids

Lattice QCD
Nuclear physics
Atomic physics
Materials science
Eigenvalues and singular values important tool

• in computational sciences
  
  Stability analysis (norm/condition number estimation)
  
  Low rank approximations (model reduction)
  
  Variance reduction in Monte Carlo methods
  
  Deflation preconditioning

• in graph analysis
  
  Graph partitioning, coloring
  
  Network analysis

• in data sciences
  
  Principal Component Analysis, Latent Semantic Indexing, Page-rank
  
  Combining with sparse approximations (sparse+low rank)
A walk through the state-of-the-art eigenvalue iterative methods

Without preconditioning, unrestarted Lanczos or Arnoldi are the optimal methods in terms of matvecs

Add preconditioning to the basic iteration $\Rightarrow$ Generalized Davidson (GD)
Like FGMRES. More expensive per iteration but very flexible

Work on a block of vectors per iteration $\Rightarrow$ block Lanczos, block GD.
More robust for multiplicities but slower convergence
A walk through the state-of-the-art eigenvalue iterative methods

Without preconditioning, unrestarted Lanczos or Arnoldi are the optimal methods in terms of matvecs

Add preconditioning to the basic iteration \(\implies\) Generalized Davidson (GD)
Like FGMRES. More expensive per iteration but very flexible

Work on a block of vectors per iteration \(\implies\) block Lanczos, block GD.
More robust for multiplicities but slower convergence

Yet,

Limited memory \(\implies\) Restarting \(\implies\) slow convergence, misconvergence

Thick restart = Implicit restart
Keeping nearby spectrum at restart important but only part of the answer
A walk through the state-of-the-art eigenvalue iterative methods

Approach it as a non-linear problem:

Newton $\Rightarrow$ Jacobi Davidson inner-outer method
Stop inner when InvIt benefits exhausted $\Rightarrow$ near optimal JDQMR

Non-linear CG $\rightarrow$ LOPCG (locally optimal restarting)

The above close to optimal for 1 extreme eigenvalue
For more eigenvalues we need block, subspace acceleration, or both
A walk through the state-of-the-art eigenvalue iterative methods

Approach it as a non-linear problem:

**Newton** $\mapsto$ **Jacobi Davidson** inner-outer method
Stop inner when InvIt benefits exhausted $\mapsto$ near optimal JDQMR

Non-linear CG $\mapsto$ LOPCG (locally optimal restarting)

The above close to optimal for 1 extreme eigenvalue
For more eigenvalues we need block, subspace acceleration, or both

**block LOPCG** $\mapsto$ **LOBPCG**
requires $block = nev$

**block GD+k = GD restarted in a LOPCG way**
Similar to limited-memory BFGS
Can use any block size $\geq 1$
Near optimal for 1 or a few extreme eigenvalues
A walk through the state-of-the-art eigenvalue iterative methods

For large $nev$ and for interior eigenproblems also much interest in:

**Polynomial filtering** where $p(A)$ is the operator in Lanczos or GD
– Reduces iteration/orthogonalization costs and parallel syncs
– However, filter tuning is an art, and matvecs increase
A walk through the state-of-the-art eigenvalue iterative methods

For large $nev$ and for interior eigenproblems also much interest in:

**Polynomial filtering** where $p(A)$ is the operator in Lanczos or GD
  - Reduces iteration/orthogonalization costs and parallel syncs
  - However, filter tuning is an art, and matvecs increase

**Contour integration**, $\oint_{\Gamma} (A - zI)^{-1}dz$, $\Gamma$ encloses the desired spectrum
  - An approximate spectral projector used with subspace iteration
  - Linear systems must be solved with complex shifts for each vector in subspace
  - Could be very efficient with direct linear solvers
  - Various levels of parallelism
  - Without direct solvers its performance worse than filtering
State-of-the-art software for Eigenproblems

- **Without preconditioning:**
  - **ARPACK** (Implicitly Restarted Arnoldi)
  - **TRLAN** (Implicitly Restarted Lanczos)
  - **FILTLAN** (Polynomially filtered unrestarted Lanczos)

- **With preconditioning, general purpose is more challenging:**
  - **Anasazi** (block GD, LOBPCG, IRTR, in Trilinos)
  - **BLOPEX** (LOBPCG)
  - **SLEPc** (JD and most major methods, extends PETSc)
  - **FEAST** (Contour integration, available in MKL)
  - **PRIMME** (block GD+k/JDQMR, most major methods)
PReconditioned Iterative MultiMethod Eigensolver

A no-shortcuts eigensolver, robust and efficient

• Full set of defaults for non experts
• Fully customizable for experts
• Near optimal for small \( nev \)
• 12 methods and their block versions available
• Dynamically chooses the best method
• Interior eigenvalues (RR/refined/harmonic)
• Parallel & HPC
• C, C++, F77, MATLAB, Python interfaces
• Float, double, complex, double complex

www.github.com/primme
PRIMME vs Anasazi for five lowest eigenpairs

Time ratios over JDqmr

- **PRIMME JDQMR**
- **PRIMME GD+k**
- **Anasazi IRTR**
- **Anasazi LOBPCG**
- **Anasazi BD**

Data points for:
- **Andrews**
- **Lap7p1M**
- **Plate33K**
- **cfd1**
- **or56f**
The SVD problem

An eigenvalue problem either on $A^T A$ or on $\begin{pmatrix} 0 & A^T \\ A & 0 \end{pmatrix}$

- Solving $A^T A$ faster but with relative error $\frac{\|A\|^2}{\sigma_i^2} \varepsilon_{mach} = \begin{cases} \varepsilon_{mach}, & \sigma_1 = \|A\| \\ \kappa(A)^2 \varepsilon_{mach}, & \sigma_N \end{cases}$

- Augmented difficult interior problem

- Lanczos Bidiagonalization (LBD) $\Leftrightarrow$ Lanczos on $A^T A$ but more accurate
  With restarting, eigenmethods on $A^T A$ still faster

- JDSVD is a preconditioned inner-outer method on the augmented.
The SVD problem: PRIMME’s two stage approach

1. Use best eigensolver on $A^T A$ up to accuracy limit
2. If further accuracy needed, continue on the augmented
   – Fine-tunes methods and their transition
   – Allows for preconditioning
   – Carries all PRIMME functionality and interfaces

Other SVD-specific software:

- **PROPACK** (implicitly restarted LBD in F77)
- **SLEPc** (thick restarted LBD in C)
- **IRLBA** (thick restarted block LBD in R)
- **JDSVD** (only in MATLAB)
Finding a few singular values

Comparing against the industry’s standard PROPACK:

PRIMME more efficient and significantly more robust
Challenges

What is the best way to compute:

1. **MANY** eigen/singular values $O(10^2 - 10^4)$
2. **INTERIOR** eigenvalues inside the spectrum
3. **MANY INTERIOR** eigenvalues

Increasingly needed in:

- Quantum chemistry (many occupied states or excited states)
- Low rank approximation (model reduction, variance reduction, embeddings)
- Computation of trace, determinant, density of states, etc
- Smallest singular values (stability analysis, condition number estimation)
Many eigenvalues challenges

Algorithms and architecture

Orthogonalization
- Cost grows as $O(nev^2N)$

Extraction method = projection and solution of projected problem
- Small basis size $\Rightarrow$ slow convergence
- Large basis size cost grows as $O(basisSize^3 + basisSize^2N)$
Many eigenvalues challenges

Algorithms and architecture

Orthogonalization
  – Cost grows as $O(n^2 N)$

Extraction method = projection and solution of projected problem
  – Small basis size ⇒ slow convergence
  – Large basis size cost grows as $O(basisSize^3 + basisSize^2 N)$

Inner-Outer
  – Reduces extraction and ortho costs
  – More inner iterations increases total “matvecs”
  – Filters or inner-outer methods?
Many eigenvalues challenges

Orthogonalization
  - Cost grows as $O(nev^2N)$

Extraction method = projection and solution of projected problem
  - Small basis size ⇒ slow convergence
  - Large basis size cost grows as $O(basisSize^3 + basisSize^2N)$

Inner-Outer

Block methods
  - Induce BLAS 3 ops in extraction and restarting
  - Optimal block size (convergence vs robustness vs GFLOPS)?
  - Allows ortho to use BLAS 3. How?
    - TSQR accurate but less efficient than SVQB
    - Compensate instead for loss of orthogonality?
  - Sparse MV is memory bound so special block MV needed
Many eigenvalues challenges

Orthogonalization
- Cost grows as $O(nev^2N)$

Extraction method = projection and solution of projected problem
- Small basis size $\Rightarrow$ slow convergence
- Large basis size cost grows as $O(basisSize^3 + basisSize^2N)$

Inner-Outer

Block methods

Communication avoiding
- Based on s-step iterative methods
- s typically cannot be very large
- Also block ortho concerns vs convergence
- Not necessary less memory accesses
Interior eigenproblem challenges

Extraction method
- Rayleigh-Ritz method fastest when it works
- Refined slower but robust
- Harmonic not easy to black box
Combination of techniques can cure some of these issues

Restarting method
- Thick restarting necessary
- Effective +k restarting is involved

Block methods
- May improve robustness at the cost of convergence speed

Preconditioning
- Use when available, but what is a good “indefinite” preconditioner?
- Use filters or inner-outer methods?
Many Interior challenges

Filtered, Inner-Outer
  – What is optimal inner degree? How about deg=2?
  – How do we pick the range to filter?
  – How do we know the #evals?

Contour integration
  – If direct solvers possible, good approach
  – With iterative linear methods, is it competitive?

Spectrum slicing
  – Iteration and ortho costs only for the slice
  – Multiple levels of parallelism
  – Load balancing of the slices (#evals and convergence rate)?
  – General technique that can be used with many methods
Results and future projections

laplacian3d_28 1000 eigs tol 1e−6

Time

10^2
10^3
10^4
10^5

MV

10^5
10^6
10^7
10^8

FEAST

JDQMR

FILTLAN

GD+k

feast 32
feast 32 ideal
GD+k
GD+k ideal
filtlan
feast 16
feast 16 ideal
feast 8
feast 8 ideal
JDQMR
JDQMR ideal
Other challenges

Very low accuracy
  – DM/ML need low accuracy (1e-2 or worse) $\implies$ Block methods needed

Very high accuracy
  – Rare, but obtaining $\|A\|_{\epsilon_{mach}}$ is demanding on the software

Randomized methods for SVD/eigenvalues
  – Plain subspace iteration
  – Randomness comes from starting block
  – Why claim more effective?
    · low accuracy requirement
    · weaker convergence criterion $\|A - VV^T A\| < \delta$
  – Why not use state-of-the-art methods with this criterion?

Nonlinear eigenvalue problems
  – Typical linearization produces standard eigenproblems of very large size
  – Specialized methods and alternatives are the focus of much research