A High-Performance Preconditioned SVD Solver for Accurately Computing Large-Scale Singular Value Problems in PRIMME

Lingfei Wu, Andreas Stathopoulos (Advisor), Eloy Romero (Advisor), College of William and Mary, Williamsburg, Virginia, USA

Introduction

❖ **The problem: find k extreme singular triplets of $A^{m\times n}$**

$$Av_i = \sigma_i u_i, \quad i = 1, 2, \ldots, k, \quad k \ll n$$

❖ **Iterative methods for computing SVD:**

❶ Hermitian eigenvalue problem on $C = A^TA$ or $C = AA^T$

❷ Hermitian eigenvalue problem on $B = [0\ A^T; A\ 0]$

❸ Lanczos bidiagonalization method (LBD)

$$A = PB_dQ^T \text{ and } B_d = X\Sigma Y^T$$

where U = PX and V = QY.

❖ **Comparison of different approaches**

Eigenmethod on C	Eigenmethod on B	LBD on A
• Fast for largest SVs • Slow for smallest SVs • Achieve accuracy of $O(\kappa(A)\|\|A\|\|\varepsilon_{mach})$	• Slow for largest SVs • Extremely slow for smallest SVs (interior eigenvalue problem) • Achieve accuracy of $O(\|\|A\|\|\varepsilon_{mach})$	• Fast for largest SVs • Similar to C but exhibits irregular convergence for smallest SVs • Achieve accuracy of $O(\|\|A\|\|\varepsilon_{mach})$

Fig. 1: Advantages and disadvantages of different approaches

State-of-The-Art SVD Solvers

❖ **Scarcity of SVD software for large-scale problems**

Software	State-of-the-art Methods	Classic Methods	Lang	MPI/ SMP	Precond-itioning	Extreme SVs	Fast Full Accuracy
PRIMME	PRIMME_SVDS	Multimethod	C	Both	Y	Y	Y
N	IRRHLB	N/A	Matlab	N	N	Y	Y
N	IRLBA	N/A	Matlab	N	N	Y	Y
N	JDSVD	N/A	Matlab	N	Y	Y	Y
N	SVDIFP	N/A	Matlab	N	Y	Y	N
SLEPc	N/A	Many	C	MPI	Y	Y	N
PROPACK	N/A	LBD	F77	SMP	N	Y	N
SVDPACK	N/A	Lanczos	F77	N	N	Y	N

Table 1: Functionalities of different SVD solvers

❖ **A clear need for a high quality SVD solver software**

➢ Implement state-of-the-art methods with massively parallelism
➢ Compute smallest singular triplets effectively and accurately
➢ Leverage preconditioning to deal with growing size and difficulty of real-world problems
➢ Flexible interface suitable for black-box and advanced usage

PRIMME_SVDS: A High-Performance SVD Solver

❖ **Our goal: solve large-scale, sparse SVD problems with unprecedented efficiency, robustness and accuracy**

❖ **Our approach: a preconditioned hybrid, two-stage SVD method on top of the state-of-the-art eigensolver PRIMME**

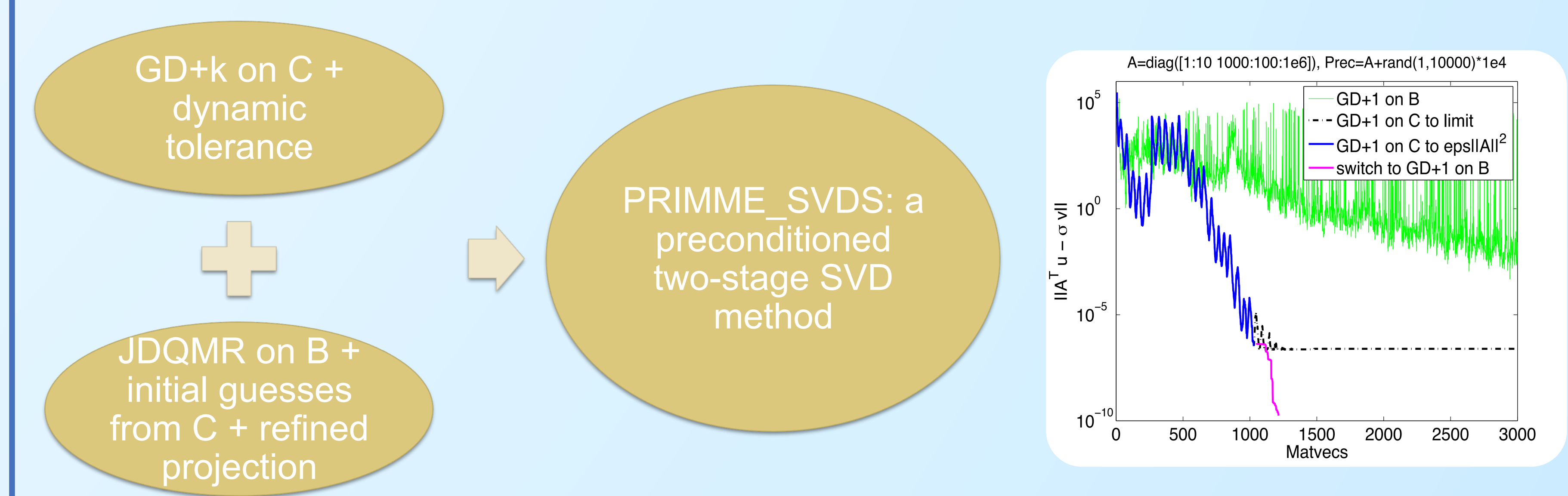

Fig. 2: Hybrid, two-stage SVD method and an example

❖ **Key components:**

➢ Stage I: fast convergence on matrix C to compute approximate eigenvalues and eigenvectors
➢ Stage II: improve accuracy by exploiting power of PRIMME and specialized refined projection on matrix B

❖ **Parallel Implementation of PRIMME_SVDS**

➢ User defined data distribution among processes
➢ User defined parallel matrix-vector and preconditioning functions
➢ User provided global summation for dot products

Operations	Kernels or Libs	Cost per iteration	Scalability
Dense algebra: MV, MM, Inner Prods, Scale	BLAS (eg, MKL, ESSL, OpenBlas, ACML)	O((m+n)*numSVs)	Good
Sparse algebra: SpMV, SpMM, Preconditioner	User defined (eg, PETSc, Trilinos, HYPRE, librsb)	O(1) calls	Application dependent
Reduction	User defined (eg, MPI_AllReduce)	O(1) calls of size O(numSVs)	Machine dependent

Table 2: Parallel characteristics of PRIMME operations

Experiments and Results

Applications	DNA	Least squares	Graph clustering	QCD
Matrix	relat9	cage15	delaunay_n24	Laplacian
Dimension	12,360,060×549,336	5,154,859	16,777,216	8,000 per proc
NNZ	38,955,420	99,199,551	100,663,202	55,760 per proc
Target	Smallest	Smallest	Largest	Largest
Preconditioning	N/A	BoomerAMG	N/A	N/A

Table 3: Dimension of the matrices and parameters used in the experiments

❖ **Experimental setup:**

➢ Compare PRIMME_SVDS against TRLBD, Krylov-Schur (KS), Generalized Davidson (GD), and Jacob-Davidson (JD) in SLEPc
➢ Seek a small number of extreme singular triplets W/ and W/O preconditioning
➢ Matrices and vectors distributed on total 1024 processes on Edison (NERSC) or 96 processes on SciClone (W&M)

❖ **Results I: methods comparison (low accuracy)**

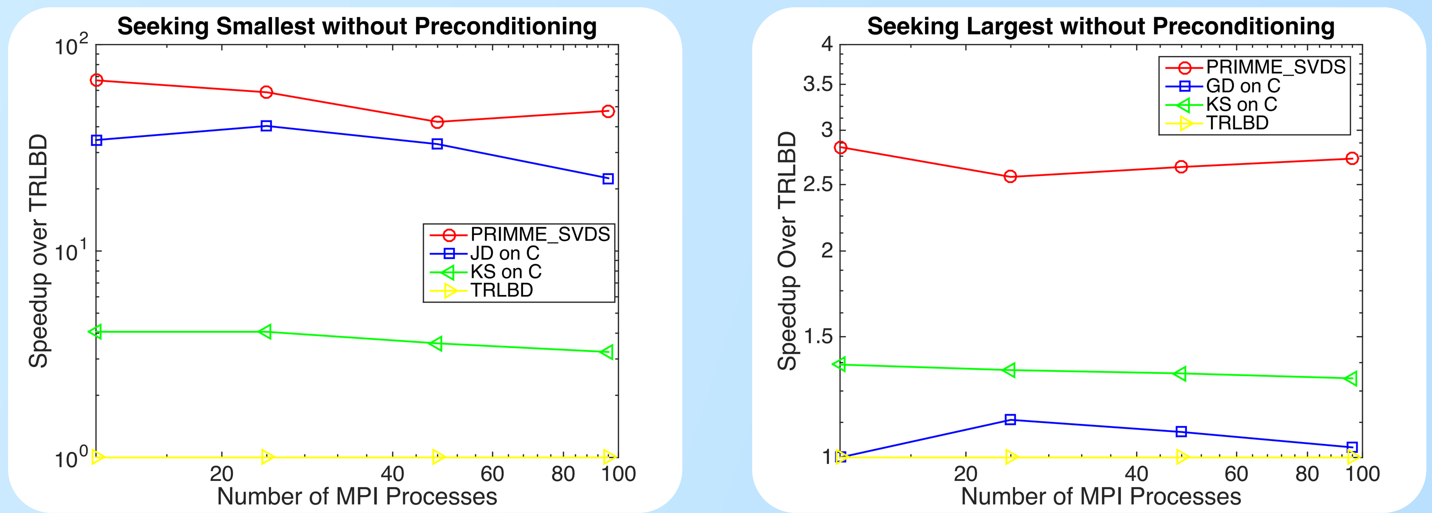

Fig. 3: Comparing different methods on relat9 and delaunay_n24

❖ **Results II: scalability performance (high accuracy)**

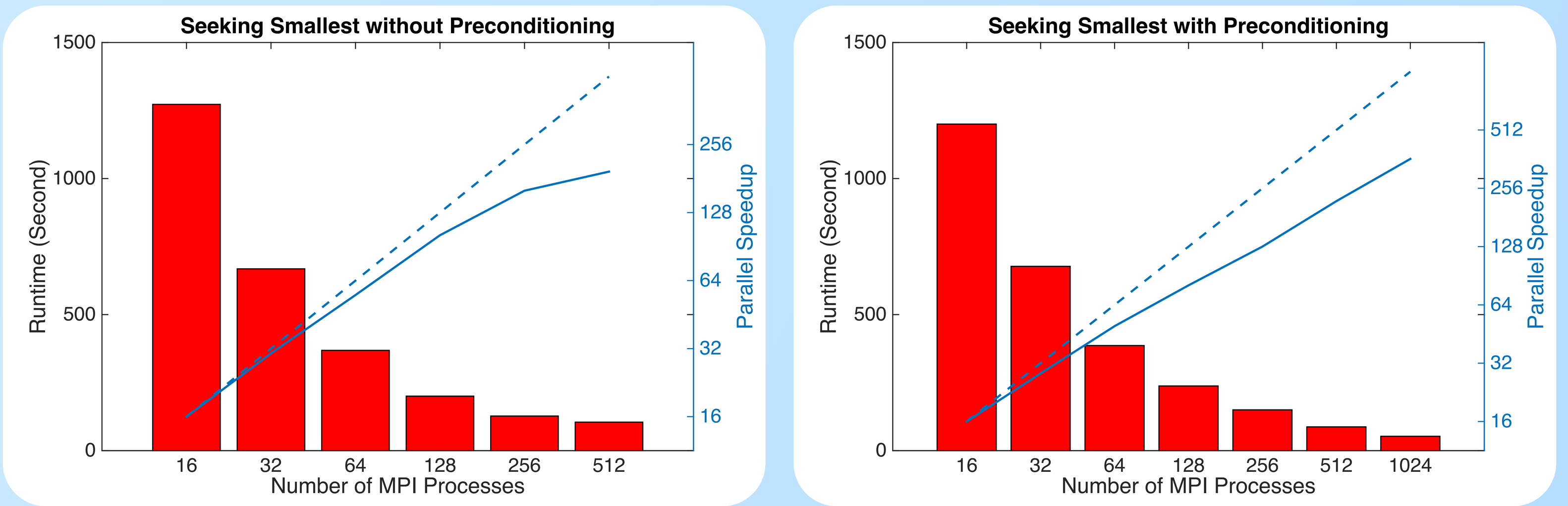

Fig. 4: Strong scaling: seek smallest on relat9 and cage15

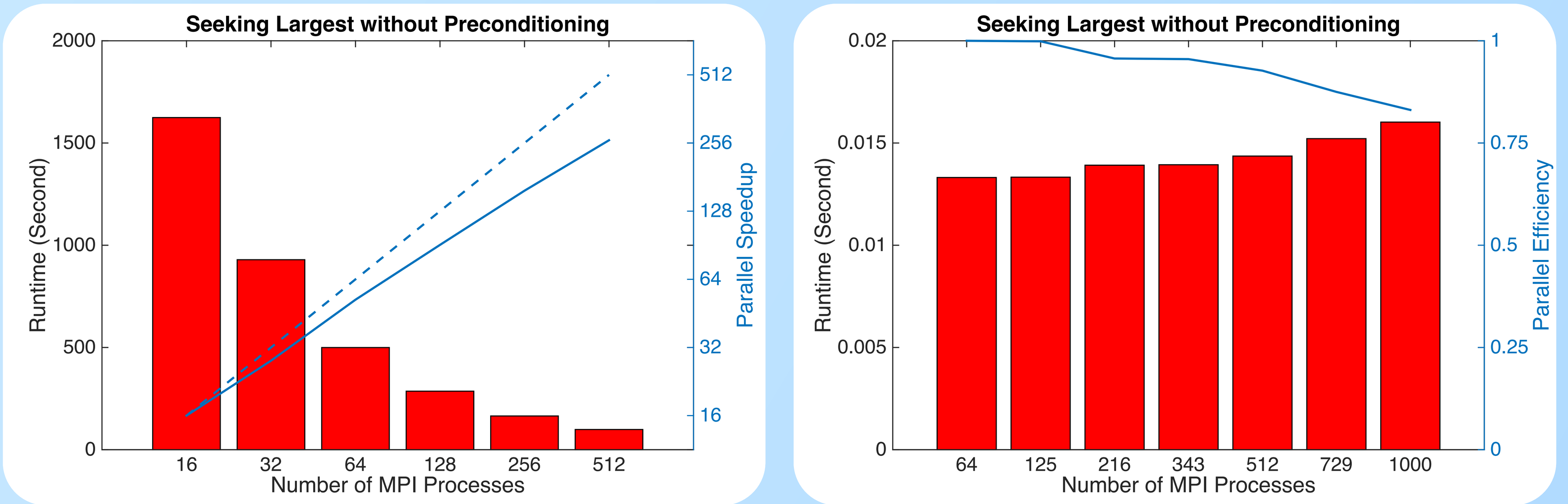

Fig. 5: Strong and Weak scaling: seek largest on delaunay_n24 and Laplacian

❖ **Highlights of PRIMME_SVDS:**

➢ Among the fastest and most robust production level software for computing a small number of singular triplets
➢ Exploits preconditioning for large-scale problems
➢ Computes efficiently smallest SVs in full accuracy
➢ Demonstrates good scalability under both strong and weak scaling even for highly sparse matrices
➢ Free software, available at: https://github.com/primme/primme

This work is supported by NSF under grants No. CCF 1218349 and ACI SI2-SSE 1440700, and by DOE under a grant No. DE-FC02-12ER41890

WILLIAM & MARY

NSF

U.S. DEPARTMENT OF ENERGY