Algorithms For Eigenvalue Optimization

文 再 文

北 京 国 际 数 学 研 究 中 心
北 京 大 学

第十届全国计算数学年会

致谢

刘歆，张寅，王晓，Michael Ulbrich，印卧涛，袁亚湘

国家自然科学基金

Outline

Optimization with orthogonality constraints:

$$\min_X \quad F(X), \quad X^\top X = I,$$

where X can be a vector or a matrix

- General Theory and Algorithms
  - Thanks: Wotao Yin
- Linear Eigenvalue Problem
  - Simple approach: concurrency and warm-start
  - Thanks: Yin Zhang, Xin Liu
- Nonlinear Eigenvalue Problem
  - Density functional theory (DFT)
    Thanks: Xin Liu, Xiao Wang, Michael Ulbrich, Yaxiang Yuan

Linear eigenvalue problem

Given a symmetric $n \times n$ real matrix A

- Eigenvalue decomposition

$$A = Q\Lambda Q^{\mathrm{T}}.$$

  - $Q \in \mathbb{R}^{n\times n}$ is orthogonal, $\Lambda \in \mathbb{R}^{n\times n}$ is diagonal.
  - Eigenvalues in descending order: $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$

- k–truncated decomposition ($k \ll n$):

$$AQ_k = Q_k\Lambda_k.$$

  - $\Lambda_k \in \mathbb{R}^{k\times k}$ contains k smallest/largest eigenvalues.
  - $Q_k \in \mathbb{R}^{n\times k}$ consists of the first/last k columns of Q.

- Trace minimization:

$$\min(\max) \quad \mathrm{tr}(X^\top AX), \text{ s.t. } X^\top X = I$$

Old? Plenty of vigor from big data analysis . . .

- A fundamental tool for many emerging optimization problems
  - First-order methods for semidefinite program
  - Low-rank matrix completion
  - Robust principal component analysis
  - Sparse principal component analysis
  - Sparse inverse covariance matrix estimation
  - Nearest correlation matrix estimation
- Various scientific and engineering applications
  - High dimensional data reduction
  - Density functional theory for electronic structure calculation
  - Nonlinear eigenvalue problems

High Dimensional Problem

- Consider

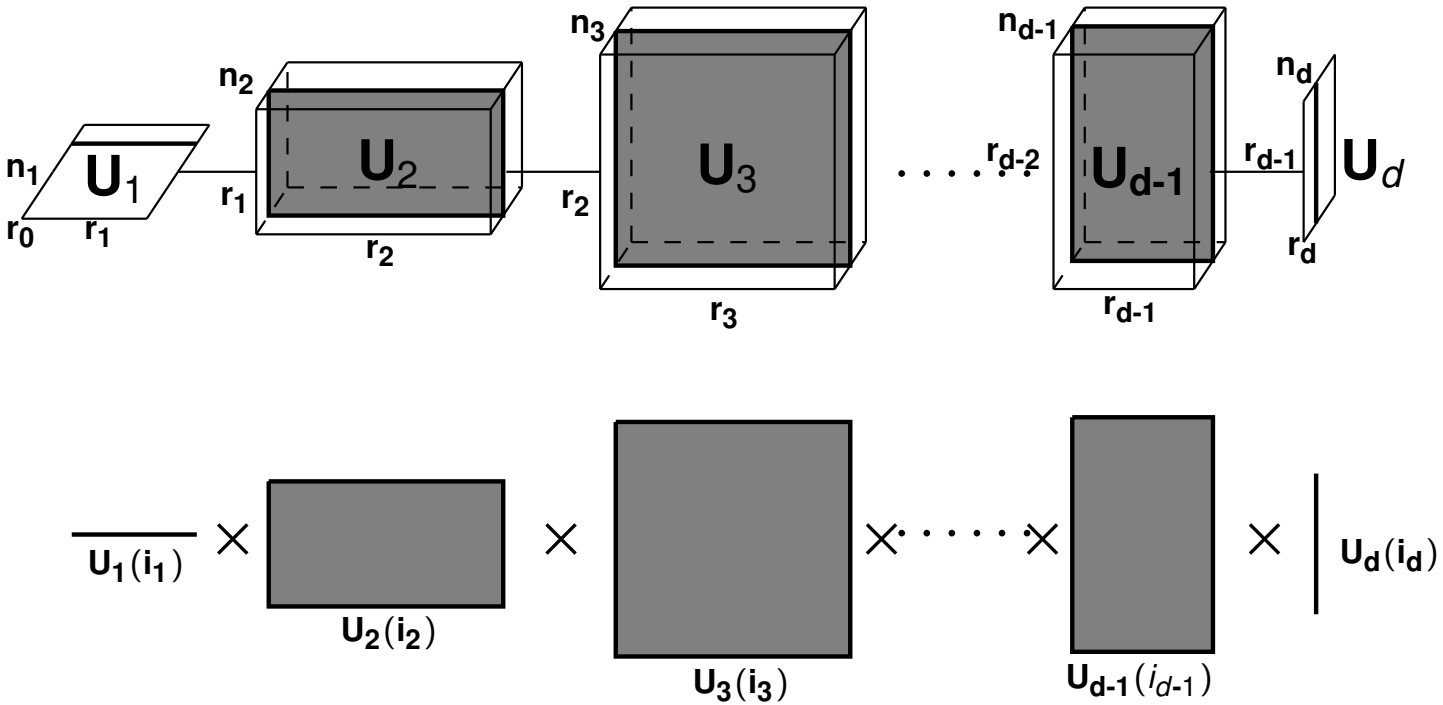
Finding p smallest eigenpairs

$$\min_{U\in\mathbb{R}^{N\times p}} \operatorname{tr}(U^\top AU), \quad \text{s.t.} \quad U^\top U = I_p,$$

- large-scale problems whose size N can be up to 10^{42}!
- Example: solving the Schrödinger equation in quantum mechanics, the discretized matrix can be as large as $n^d \times n^d$, where n stands for the discretized point in one dimension and d represents the dimension.
- Even the storage of the full matrix may be impossible and it is far beyond the tractability of classic eigensolvers.

Tensor-Train (TT) Format

- Graphical representation



Nonlinear Eigenvalue Problems in DFT

- Kohn-Sham total energy minimization:

$$\min_{X^*X=I} E_{KS}(X) := E_{kinetic}(X) + E_{ion}(X) + E_{Hartree}(X) + E_{xc}(X),$$

where

$$\begin{aligned} E_{kinetic}(X) &= \frac{1}{2}\mathrm{tr}(X^* L X) \\ E_{ion}(X) &= \mathrm{tr}(X^* V_{ion} X) \\ E_{Hartree}(X) &= \frac{1}{2}\rho(X)^\top L^\dagger \rho(X) \\ E_{xc}(X) &= \rho(X)^\top \mu_{xc}(\rho(X)) \\ \rho(X) &= \mathrm{diag}(XX^*) \end{aligned}$$

- Kohn-Sham equation:

$$\begin{aligned} H(X)X &= X\Lambda \\ X^*X &= I \end{aligned}$$

The Orthogonality Constrained Problems

Consider $X \in \mathbb{R}^{n\times p}$ and

$$\min F(X), \quad \text{subject to} \quad X^\top X = I$$

If $p = 1$, X reduces to $x \in \mathbb{R}^n$,

$$\|x\|_2 = 1.$$

Why is the problem interesting?

- it is expensive to keep constraints feasible;
- non-convex, leading to possibly local minima.
- it has many applications;

Motivation

Consider

$$\min F(X), \quad \text{subject to} \quad X^\top X = I.$$

The idea is like Crank-Nicolson: at iteration k

$$\boxed{X_{k+1} \leftarrow \mathrm{Orth} \leftarrow X_k + \tau \mathrm{Proj}(-\nabla F_k)}$$

$$\Updownarrow$$

$$\boxed{X_{k+1} \leftarrow \mathrm{Orth} \leftarrow X_k + \tau H_k X_k}$$

$$\Downarrow$$

$$\boxed{X_{k+1} \leftarrow \text{solve } Y = X_k + \tfrac{\tau}{2} H_k (X_k + Y)}$$

An Update Formula preserving Feasibility

Given X_k and $G_k = \nabla F(X_k)$, we let

$$H = X_k G_k^\top - G_k X_k^\top$$

and solve

$$Y = X + \frac{\tau}{2} H(X + Y)$$

for $Y(\tau)$. Update

$$X_{k+1} \leftarrow Y(\tau) = \left(I - \frac{\tau}{2} H\right)^{-1} \left(I + \frac{\tau}{2} H\right) X_k,$$

where τ is a step size.

Properties of $Y(\tau)$:

1. $Y(\tau)^\top Y(\tau) = X_k^\top X_k$;
2. $Y(\tau)$ is a descent curve;
3. $Y(\tau)$ has fast implementations.

$Y(\tau)$ is a descent curve

- $Y'(0) =$ negative projected gradient.

$$Y'(0) = HX_k = \mathrm{Proj}^c(-\nabla F_k; X_k).$$

Proj^c is projection to the tangent of $\{X : X^\top X = I\}$ under the *canonical metric*:

$$\langle Y, Z\rangle_c = \mathrm{tr}(Y^\top(I - \frac{1}{2}XX^\top)Z), \quad Y, Z \in T_X.$$

- To get projection under the *Euclidean metric*: $\langle Y, Z\rangle_e = \mathrm{tr}(Y^\top Z)$, use

$$H = XG_k^\top(I - \frac{1}{2}X_kX_k^\top) - (I - \frac{1}{2}X_kX_k^\top)G_kX^\top.$$

- Generally, given any tangent direction $D \in T_X$ (from CG, Newton, quasi-Newton),

$$H = (I - \frac{1}{2}XX^\top)DX^\top - XD^\top(I - \frac{1}{2}XX^\top)$$

leads to $Y'(0) = D$.

Discretized Kohn-Sham Formulation

- Goal: find ground state energy/density by minimizing E_{KS}.
- Finite dimensional problem:

$$\min_{X^*X=I} E_{KS}(X) := E_{kinetic}(X) + E_{ion}(X) + E_{Hartree}(X) + E_{xc}(X),$$

where $X \in \mathbf{C}^{n\times p}$,

$$\begin{aligned}
E_{kinetic}(X) &= \tfrac{1}{2}\mathrm{tr}(X^*LX) \\
E_{ion}(X) &= \mathrm{tr}(X^*V_{ion}X) + \sum_i \sum_l |x_i^* w_l|^2 \\
E_{Hartree}(X) &= \tfrac{1}{2}\rho(X)^\top L^\dagger \rho(X) \\
E_{xc}(X) &= e^\top \epsilon_{xc}(\rho(X)), \quad e = (1,\ldots,1)^\top \\
\rho(X) &= \mathrm{diag}(XX^*) = (\sum_{j=1}^{p} |x_{ij}|^2)_{1\le i\le n}
\end{aligned}$$

KKT Conditions

- Lagrange function: $\mathcal{L}(X,\Lambda) = E_{KS}(X) - \frac{1}{2}\mathrm{tr}(\Lambda(X^*X - I))$
- First-order optimality conditions:

$$\begin{cases} \nabla_X \mathcal{L}(X,\Lambda) = 0. \\ X^*X = I, \end{cases} \Longrightarrow \begin{cases} H(\rho(X))X = X\Lambda, \\ X^*X = I. \end{cases}$$

- Kohn-Sham Equation:

$$\begin{cases} H(\rho(X))X = X\Lambda, \\ X^*X = I. \end{cases}$$

where Λ is the smallest p eigenvalues of $H(X)$.

- Kohn-Sham Hamiltonian:

$$H(\rho(X)) := \frac{1}{2}L + V_{ion} + \sum_l w_l w_l^* + \mathrm{diag}\left(\Re(L^\dagger)\rho(X) + \partial_\rho \epsilon_{xc}(\rho(X))^\top e\right)$$

Formulating the KS Equation as a Fixed Point Map

- Nonlinear equations with respect to ρ as

$$\rho = \mathrm{diag}(X(\rho)X(\rho)^{\mathrm{T}}).$$

- X is determined by the eigenvalue problem:

$$\begin{cases} \hat{H}(\rho)X = X\Lambda, \\ X^{\mathrm{T}}X = I, \end{cases}$$

- the Hamiltonian matrix

$$\hat{H}(\rho) := \frac{1}{2}L + V_{ion} + \mathrm{Diag}(L^{\dagger}\rho) + \mathrm{Diag}(\mu_{xc}(\rho)^{\mathrm{T}}e)$$

Formulating the KS Equation as a Fixed Point Map

- The Hamiltonian matrix

$$H(V) := \frac{1}{2}L + V_{ion} + \mathrm{Diag}(V)$$

- The potential

$$V := \mathcal{V}(\rho) = L^{\dagger}\rho + \mu_{xc}(\rho)^{\mathrm{T}}e$$

- Nonlinear equations with respect to

$$\begin{cases} V = \mathcal{V}(F_{\phi}(V)), \\ F_{\phi}(V) = \mathrm{diag}(X(V)X(V)^{\mathrm{T}}). \end{cases}$$

The Jacobian of the Fixed Point Maps

- Let $\{\lambda_i(V), q_i(V)\}$ be the eigenpairs of $H(V)$:

$$\lambda_1(V) \le \ldots \le \lambda_p(V) \le \lambda_{p+1}(V) \le \ldots \le \lambda_n(V).$$

- The eigenvalue decomposition of $H(V)$:

$$H(V) = Q(V)\Pi(V)Q(V)^{\mathrm{T}},$$

- The function $F_\phi(V)$ in (15) is equivalent to

$$F_\phi(V) = \mathrm{diag}(Q(V)\phi(\Pi(V))Q(V)^{\mathrm{T}}),$$

where $\phi(\Pi) = \mathrm{Diag}(\phi(\lambda_1(V)), \phi(\lambda_2(V)), \ldots, \phi(\lambda_n(V)))$ and

$$\phi(t) := \begin{cases} 1 & \text{for } t \le \frac{\lambda_p(V)+\lambda_{p+1}(V)}{2}, \\ 0 & \text{for } t > \frac{\lambda_p(V)+\lambda_{p+1}(V)}{2}. \end{cases}$$

The Jacobian of the Fixed Point Maps

- Suppose that $\lambda_{p+1}(V) > \lambda_p(V)$. Then the directional derivative of $F_\phi(V)$ at V is

$$\partial_V F_\phi(V)[Z] = \operatorname{diag}\Big(Q(V)\Big(g_\phi(\Pi(V)) \circ \Big(Q(V)^{\mathrm{T}} \operatorname{Diag}(z)\, Q(V)\Big)\Big) Q(V)^{\mathrm{T}}\Big),$$

  where $g_\phi(\Pi(V)) \in \mathbb{R}^{n\times n}$ is defined as

$$(g_\phi(\Pi(V)))_{ij} = \begin{cases} \frac{1}{\lambda_i(V)-\lambda_j(V)} & \text{if } i \in \alpha_k,\ j \in \alpha_l,\ k \le r_p(V),\ l > r_p(V), \\ \frac{-1}{\lambda_i(V)-\lambda_j(V)} & \text{if } i \in \alpha_k,\ j \in \alpha_l,\ k > r_p(V),\ l \le r_p(V), \\ 0 & \text{otherwise.} \end{cases}$$

- The Jacobian of $\mathcal{V}(F_\phi(V))$ at V is

$$\partial_V \mathcal{V}(F_\phi(V))[Z] = J(F_\phi(V))\partial_V F_\phi(V)[Z], \quad \text{for all } Z \in \mathbb{R}^n.$$

Convergence of the SCF iteration

- SCF recursively computes:

$$H(V^i)X(V^{i+1}) = X(V^{i+1})\Lambda(V^{i+1}),$$
$$X(V^{i+1})^{\mathrm{T}}X(V^{i+1}) = I,$$

  and then the potential is updated as

$$V^{i+1} = \mathcal{V}(F_\phi(V^i)).$$

- The simple mixing scheme replaces it by updating

$$V^{i+1} = V^i - \alpha(V^i - \mathcal{V}(F_\phi(V^i)))$$

Convergence of the SCF iteration

- Assumption: the second-order derivatives of $\epsilon_{xc}(\rho)$:

$$\|J(\rho)\|_2 = \left\| L^\dagger + \frac{\partial \mu_{xc}(\rho)}{\partial \rho} e \right\|_2 \leq \theta, \qquad \text{for all } \rho \in \mathbb{R}^n.$$

- δ: the eigenvalue gap between λ_p and λ_{p+1}
- It holds

$$\|\partial_V F_\phi(V)\|_2 \leq \frac{1}{\delta} \quad \text{and} \quad \|\partial_V \mathcal{V}(F_\phi(V))\|_2 \leq \frac{\theta}{\delta}.$$

Global Convergence of the SCF iteration

Convergence from any starting point:

- Suppose that the eigenvalue gap satisfies

$$b_1 := 1 - \frac{\theta}{\delta} > 0.$$

Let $\{V^i\}$ be generated by the simple mixing scheme using

$$0 < \alpha < \frac{2}{2 - b_1}.$$

Then $\{V^i\}$ converges to a solution of the KS equation with linear convergence rate no more than $|1 - \alpha| + \alpha(1 - b_1)$.

- Proof:

$$\|(1 - \alpha)I + \alpha \partial_V \mathcal{V}(F_\phi(V^i))\|_2 < 1$$

Numerical Verification

- Experiments using KSSOLV
- planewave discretization: $L = \frac{1}{N} F^* \mathrm{Diag}(d_g) F$
- exchange-correlation: the Perdew & Zunger formula
- ratio1: $\frac{\|J(F_\phi(V))\|_2}{\delta}$; ratio2: $\frac{-\min\{0, \lambda_{\min}(J(F_\phi(V)))\}}{\delta}$

Table: SCF using the simple mixing scheme with $\alpha = 0.5$

name	(n_1, n_2, n_3)	n	p	iter	res	d_g	$\mathrm{diag}\left(\frac{\partial \mu_{xc}(\rho)}{\partial \rho} e\right)$	ratio1	ratio2
c2h6	(32, 32, 32)	2103	7	17	1.14e-06	(1.97e-01, 4.99e+01)	(5.97e-01, 1.11e+04)	1.2860	1.2860
co2	(32, 32, 32)	2103	8	22	9.98e-08	(1.97e-01, 4.99e+01)	(3.83e-01, 1.85e+04)	2.7901	2.7901
h2o	(32, 32, 32)	2103	4	17	1.31e-06	(1.97e-01, 4.99e+01)	(4.02e-01, 1.87e+05)	33.7847	33.7847
hnco	(32, 32, 32)	2103	8	22	1.81e-07	(1.97e-01, 4.99e+01)	(3.62e-01, 1.05e+04)	4.1224	4.1223
nic	(16, 16, 16)	251	7	20	2.03e-07	(7.90e-01, 4.90e+01)	(5.21e-01, 4.26e+00)	0.0067	0.0047
si2h4	(32, 32, 32)	2103	6	18	1.69e-06	(1.97e-01, 4.99e+01)	(5.60e-01, 4.67e+03)	0.9864	0.9864
sih4	(32, 32, 32)	2103	4	16	1.12e-06	(1.97e-01, 4.99e+01)	(8.92e-01, 2.32e+03)	0.2954	0.2954

Linear eigenvalue problem is
a fundamental, classic, and well-studied problem

Why block algorithms now?

2 Advantages for Block Methods

Parallel Scalability

Warm-start capacity

Two algorithms in one framework

Task: Given large sparse $A = A^T \in \mathbb{R}^{n\times n}$, compute k largest eigenpairs (q_j, λ_j), $j = 1, \cdots, k$ for "large" $k \ll n$.

Our Framework:

1. A block method for subspace update
2. Augmented RR (ARR) projection

2 Block Method Variants for SU:

- Multi-power method
- Gauss Newton method

Acceleration: replace A by $\rho(A)$

Block Methods

Computer architectures have evolved in the last 20 years.
Number of cores/processors: ⇗.

Trends of "Relative Importance"

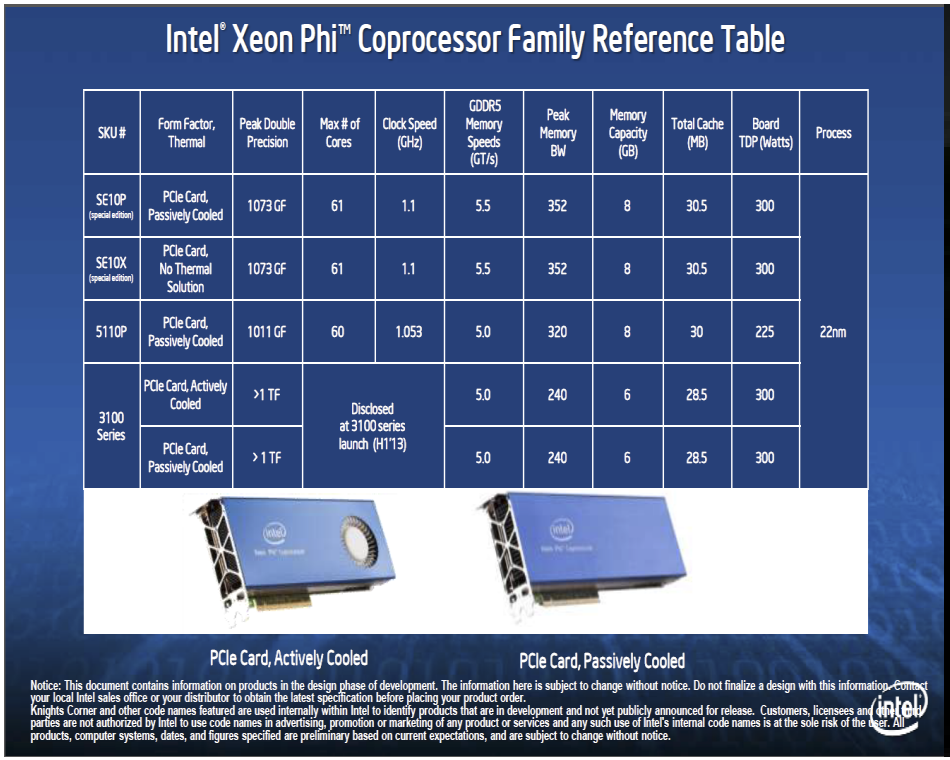
- cost of communications: ⇗
- cost of arithmetic operations: ⇘
- concurrency in algorithms: ⇗
- traditional complexity (i.e., SpMV): ⇘
- Block method (SpMB) favorability: ⇗

Classic Block Method: Subspace Iteration (power method)

$$X = orth(AX)$$

More recently: LOBPCG, Feast, LMSVD, ... , still on-going.

Computer architectures: CPU vs. GPU



Intel® Xeon Phi™ Coprocessor Family Reference Table

SKU #	Form Factor, Thermal	Peak Double Precision	Max # of Cores	Clock Speed (GHz)	GDDR5 Memory Speeds (GT/s)	Peak Memory BW	Memory Capacity (GB)	Total Cache (MB)	Board TDP (Watts)	Process
SE10P (special edition)	PCIe Card, Passively Cooled	1073 GF	61	1.1	5.5	352	8	30.5	300	22nm
SE10X (special edition)	PCIe Card, No Thermal Solution	1073 GF	61	1.1	5.5	352	8	30.5	300	
5110P	PCIe Card, Passively Cooled	1011 GF	60	1.053	5.0	320	8	30	225	
3100 Series	PCIe Card, Actively Cooled	>1 TF	Disclosed at 3100 series launch (H1'13)		5.0	240	6	28.5	300	
	PCIe Card, Passively Cooled	> 1 TF			5.0	240	6	28.5	300	

PCIe Card, Actively Cooled

PCIe Card, Passively Cooled

Notice: This document contains information on products in the design phase of development. The information here is subject to change without notice. Do not finalize a design with this information. Contact your local Intel sales office or your distributor to obtain the latest specification before placing your product order.
Knights Corner and other code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user. All products, computer systems, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.

March, 2015:

- Intel Xeon Processor E7-8880 v2, 15 cores
- Intel Xeon Phi Coprocessor 7120A, 61 cores
- Nivdia Tesla K80 (2496*2 GPUs) 4992 cores

Augmented Rayleigh-Ritz Projection

ARR Procedure: $p \geq 0$

1. Input $X \in \mathbb{R}^{n \times k}$
2. Augmentation: $Y = [X \;\; AX \;\; \cdots \;\; A^p X]$
3. Orthogonalization: $Y = \mathrm{qr}(Y, 0)$
4. Call $\mathrm{eig}(Y^{\mathrm{T}} A Y)$ to obtain $k + pk$ Ritz pairs
5. Output the first k Ritz pairs (or more)

Observations:

- $p = 0 \;\; \rightarrow \;\;$ standard RR
- size of ARR is $p + 1$ times larger
- default: $p = 1$ ($n \times 2k$ storage)

A Measure of Accuracy

Recall $A = Q\Lambda Q^T$, $Q^T Q = I$ and $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$.

For $X \in \mathbb{R}^{n\times k}$, how close is $\mathrm{span}(X)$ to $\mathrm{span}(q_1, \cdots, q_k)$?

Examine sizes of projections: $\|q_j q_j^T X\| = \|q_j^T X\|$.

Definition:

$$\delta_k(X) \triangleq \frac{\max_{j>k} \|q_j^T X\|}{\min_{j\leq k} \|q_j^T X\|}$$

The smaller $\delta_k(X)$ is, the better X is.

Let $Y = \mathrm{ARR}(X)$ be the output of ARR for given X.

- How good can Y be?
- How small is $\delta_k(Y)/\delta_k(X)$?

Accelerated Power Iteration

Let $X = \rho(A)^q X_0$ and $|\rho(\lambda_1)| \geq \cdots \geq |\rho(\lambda_{k+1+pk})| \geq$ the rest.

Theorem

There exist constants C_p and C'_p such that

$$\delta_k(Y) \leq C_p \left| \frac{\rho(\lambda_{k+1+pk})}{\rho(\lambda_k)} \right|^q. \qquad \frac{\delta_k(Y)}{\delta_k(X)} \leq C'_p \left| \frac{\rho(\lambda_{k+1+pk})}{\rho(\lambda_{k+1})} \right|^q$$

Special case: $p = 0$, $\rho(t) = t \implies \delta_k(Y) \leq C_0 \left(\lambda_{k+1}/\lambda_k\right)^q$

When is a single ARR sufficient (say, for $p = 2$)?

$$C_2 = O(10^9), \ \left| \frac{\rho(\lambda_{3k+1})}{\rho(\lambda_k)} \right|^q = O(10^{-16}) \implies \delta_k(Y) \leq O(10^{-7})$$

Single ARR is realizable in practice for error $\leq O(10^{-7})$ if:

(i) spectrum has a "regular" decay, (ii) iteration stops at a right time.

Subspace Update

Multi-Power Method (MPM): Input $X \in \mathbb{R}^{n\times k}$

- $x_j = \rho(A)x_j/\|x_j\|, \quad j = 1, \cdots, k$
- Main stopping rule: $cond(X) \geq 1/\epsilon$ (machine eps)
  Welcome rank loss (instead of keeping $cond(X)$ small)

Gauss-Newton (for GN): Input $X \in \mathbb{R}^{n\times k}$

- $Y = X(X^{\mathrm{T}}X)^{-1}; X = \rho(A)Y - X(Y^{\mathrm{T}}\rho(A)Y - I)/2$
- Other things remain the same (in trials so far)

Under ARR, we constructed 2 block methods: MPM & GN

SU step is close to "embarrassingly parallel" for MPM.

Sym. Low-Rank Product Approximation (GN)

Nonlinear Least Squares:

$$X^* = \underset{X \in \mathbb{R}^{n\times k}}{\operatorname{argmin}} \|XX^{\mathrm{T}} - A\|_{\mathrm{F}}^2.$$

Then X^* spans the k-D principal invariant subspace of A ($\lambda_k > 0$):

$$\operatorname{span}(X^*) = \operatorname{span}\{q_1, q_2, \ldots, q_k\},$$

Unit-Step GN Algorithm

Initialize $X \in \mathbb{R}^{n\times k}$. While not “converged", do

1. $Y = X(X^{\mathrm{T}}X)^{-1}$
2. $X = AY - X(Y^T AY - I)/2$

Convergence results given in X. Liu, Z. Wen and YZ.

Algorithm Sketch： Arrabit

Initialize random $X \in \mathbb{R}^{n\times k}$

1. Until inner-criteria are met,
   - do MPM or GN iterations
2. Do ARR to compute Ritz pairs (x_j, μ_j).
3. If maxres $\leq$ tol, output (x_j, μ_j) and exit
4. Do possible adjustments. Go to Step 1

$$\text{maxres} = \max_{1\leq j\leq k} \frac{\|Ax_j - x_j\mu_j\|}{\max(1, |\mu_j|)}$$

Inner-criterion: Keeps iterating until losing rank (rcond$(X) > 1/\epsilon$)

Possible adjustments: deflation, p, deg$(\rho(A))$, interval, etc.

“Large-Scale” Experiment Environment

Computing Platform

- A single node at Edison, a Cray XC30 supercomputer at NERSC in Berkeley. Two 12-core Intel “Ivy Bridge” processors at 2.4 GHz, 64 GB shared memory.

Required Accuracy: 10^{-6} or 10^{-12}

Tested Solvers

- MATLAB EIGS — Lanczos-based (ARPACK, Sorensen et.al.)
- FEAST — Contour integral (Polizzi). Finds all λ_j's in given $[a, b]$. Current Fortran version in Intel MKL library “`mexed`” into Matlab
- MPM and GN — our multi-power and GN method with ARR

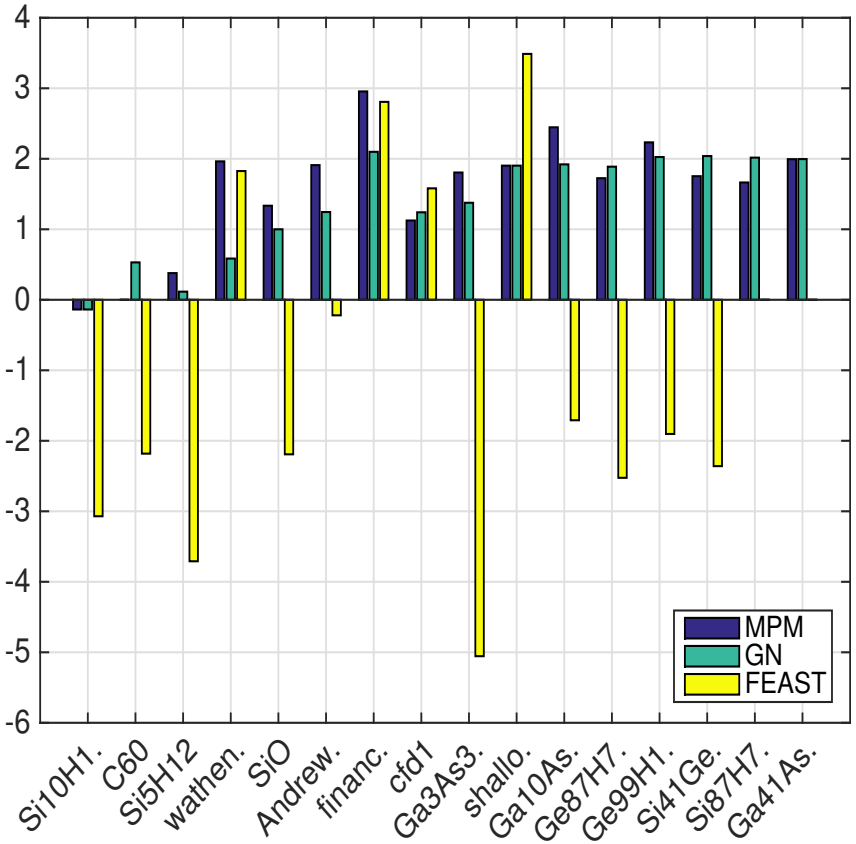
U of F Sparse Matrix Collection

Table: Information of 16 Test Matrices

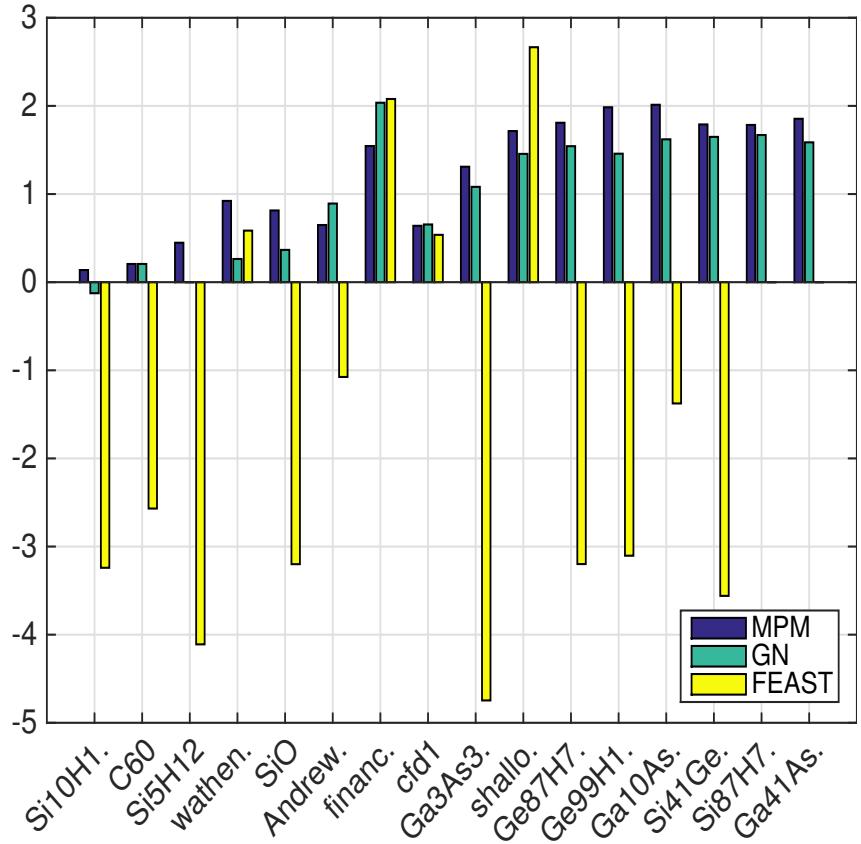
matrix name	n	k	$nnz(A)$	density of A	nnz(L)	density of L	time
Andrews	60000	600	760154	0.021%	117039940	6.502%	7.18
C60	17576	176	407204	0.132%	34144169	22.105%	1.62
cfd1	70656	707	1825580	0.037%	35877440	1.437%	1.81
finance	74752	748	596992	0.011%	2837714	0.102%	0.28
Ga10As10H30	113081	1000	6115633	0.048%	1562547805	24.439%	127.12
Ga3As3H12	61349	613	5970947	0.159%	596645077	31.705%	42.00
shallow_water1s	81920	819	327680	0.005%	2357535	0.070%	0.21
Si10H16	17077	171	875923	0.300%	56103003	38.474%	2.60
Si5H12	19896	199	738598	0.187%	78918573	39.871%	3.80
SiO	33401	334	1317655	0.118%	186085449	33.359%	10.01
wathen100	30401	304	471601	0.051%	1490209	0.322%	0.32
Ge87H76	112985	1000	7892195	0.062%	1403571238	21.990%	109.64
Ge99H100	112985	1000	8451395	0.066%	1477089634	23.141%	120.08
Si41Ge41H72	185639	1000	15011265	0.044%	3457063398	20.063%	358.53
Si87H76	240369	1000	10661631	0.018%	5568995364	19.277%	1499.80
Ga41As41H72	268096	1000	18488476	0.026%	6998257446	19.473%	2498.43

Summary (I): k largest eigenpairs

$\log_2(time_{eigs}/time)$



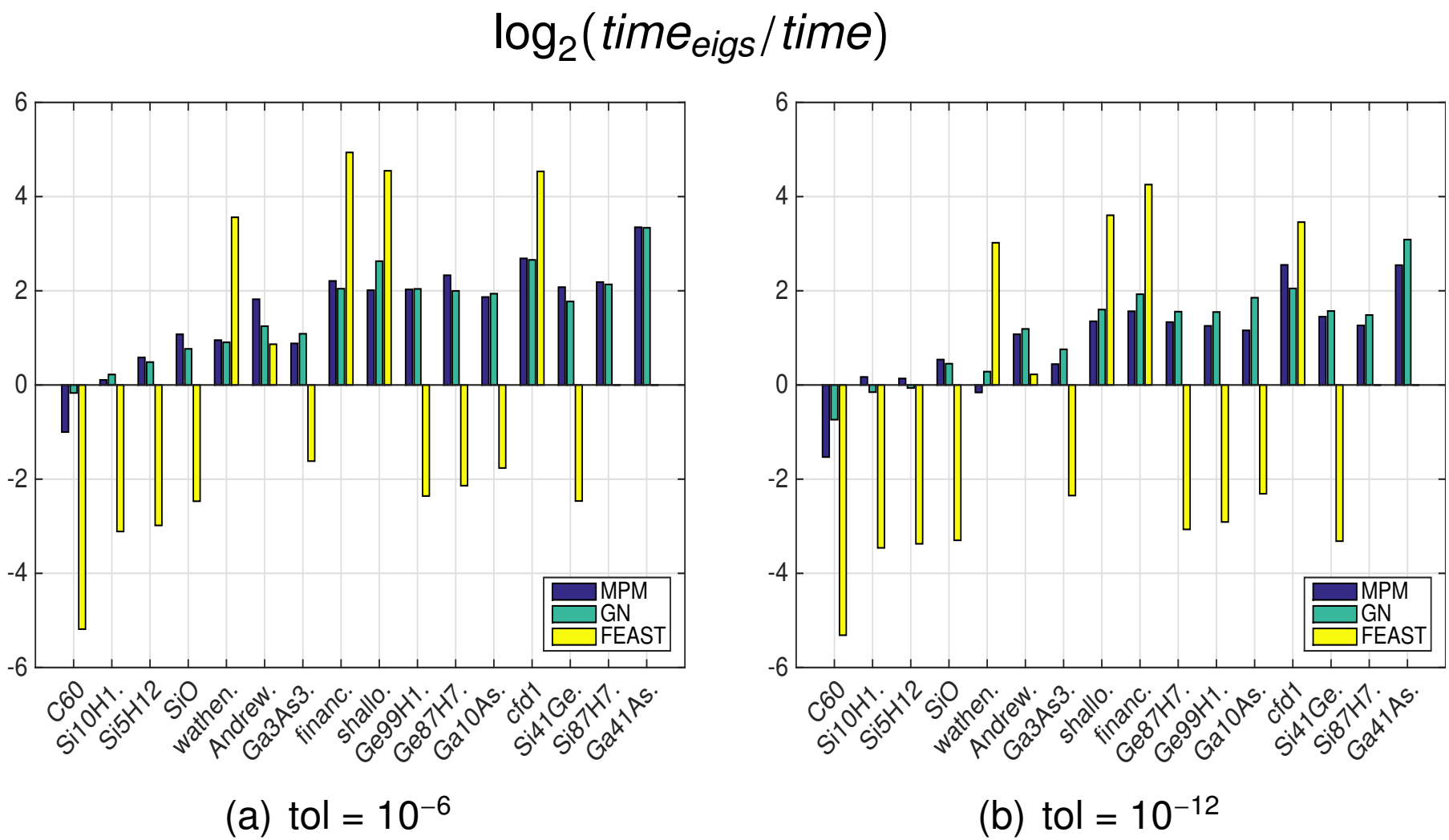
(a) tol = 10^{-6}



(b) tol = 10^{-12}

Figure: 16 Problems in ascending order of EIGS time

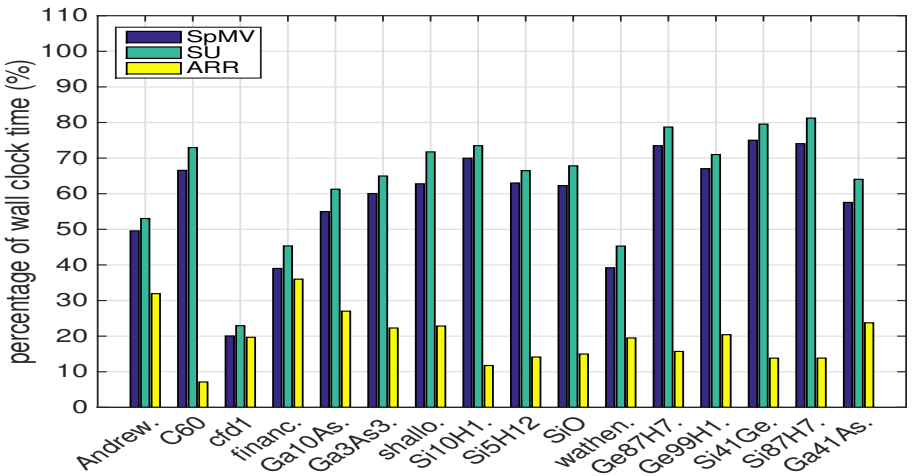
Summary (II): k smallest eigenpairs



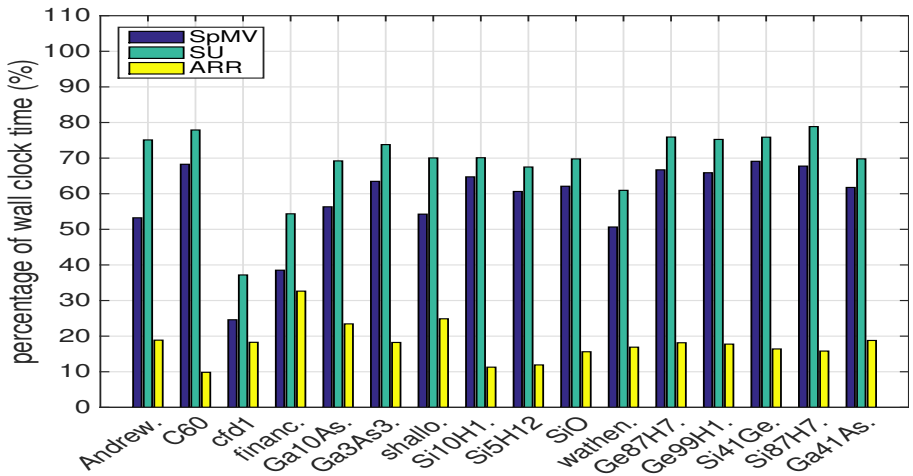
(a) tol = 10^{-6} (b) tol = 10^{-12}

Figure: 16 Problems in ascending order of EIGS time

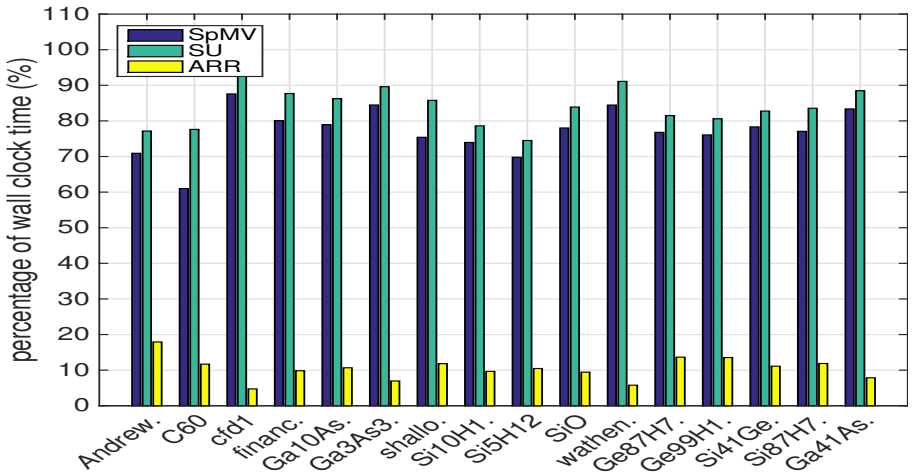
Comparison of timing among SpMV, SU and ARR



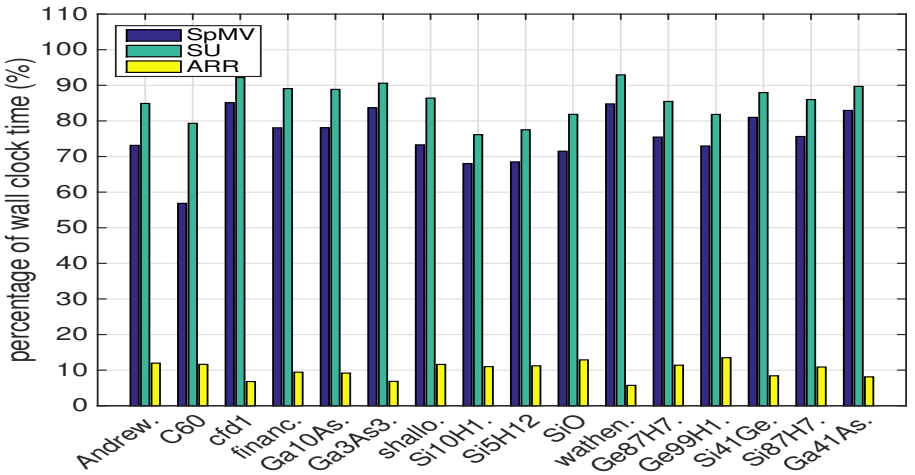
(a) $\mathrm{tol} = 10^{-6}$, MPM, k largest eigenpairs



(b) $\mathrm{tol} = 10^{-12}$, MPM, k largest eigenpairs



(c) $\mathrm{tol} = 10^{-6}$, MPM, k smallest eigenpairs



(d) $\mathrm{tol} = 10^{-12}$, MPM, k smallest eigenpairs

Contact Information

Many Thanks For Your Attention!

- Looking for Ph.D students and Postdoc
- `http://bicmr.pku.edu.cn/~wenzw`
- E-mail: wenzw@pku.edu.cn
- Office phone: 86-10-62744125