Iterative Methods for Solving Large-scale Eigenvalue Problems

Chao Yang

Lawrence Berkeley National Laboratory, Berkeley, CA

November 19, 2015

Outline

- Krylov subspace methods
- Jacobi-Davidson
- Locally optimal conjugate gradient

Matrix Eigenvalue Problem is Nonlinear

- Large-scale eigenvalue problem

$$Ax = \lambda x \quad \text{or} \quad Ax = \lambda Bx$$

  - A, B large, sparse, or structured
  - $y \leftarrow Ax$ and $y \leftarrow Bx$ can be computed efficiently
- Set of nonlinear equations in x, since $\lambda = x^T Ax / x^T Bx$.
- When A are symmetric, B positive definite, it is an optimization problem

$$\lambda_1 = \min_{x^T Bx = 1} \operatorname{trace}(x^T Ax)$$

Power iteration

Algorithm:

- Pick a starting vector v_0
- while convergence not reached
  - $w \leftarrow Av_0$;
  - $v_0 \leftarrow w/\|w\|$;

Convergence:

- linear convergence to the largest (magnitude) eigenvalue;
- rate of convergence is λ_2/λ_1;

Variations:

- To get the smallest eigenvalue, apply power iteration to A^{-1} (inverse iteration);
- To get eigenvalue closest to a target σ, apply the power iteration to $(A - \sigma I)^{-1}$;
- Rayleigh-Quotient iteration

Krylov subspace method

- Produces an orthonormal basis $V_m = \{v_1, v_2, ..., v_m\}$ associated with a *Krylov Subspace*

$$\mathcal{K}(A, v_1; m) = \text{span}\{v_1, Av_1, ..., A^m v_1\}$$

- A single step (Gram-Schmidt):

$$f_j \leftarrow (I - V_j V_j^H) A v_j; \quad v_{j+1} \leftarrow f_j / \|f_j\|;$$

- After m steps:

$$AV_m = V_m H_m + f_m e_m^H; \quad V_m^H V_m = I_m; \quad V_m^H f_m = 0;$$

The Rayleigh-Ritz procedure

- Find (θ_i, y_i) such that

$$V_m^H(AV_m y - \theta V_m y) = 0 \quad (\textit{Galerkin} \text{ condition})$$

- Equivalent to solving

$$H_m y_i = \theta_i y_i, \quad \text{where} \quad H_m = V_m^H A V_m.$$

- Approximation to

$$\begin{array}{ll} \text{eigenvalue} & \theta_i \quad (\text{Ritz value}) \\ \text{eigenvector} & z_i = V_m y_i \quad (\text{Ritz vector}) \end{array}$$

Checking Convergence

- Let $z = V_m y$, where $H_m y = \theta y$;
- Residual norm

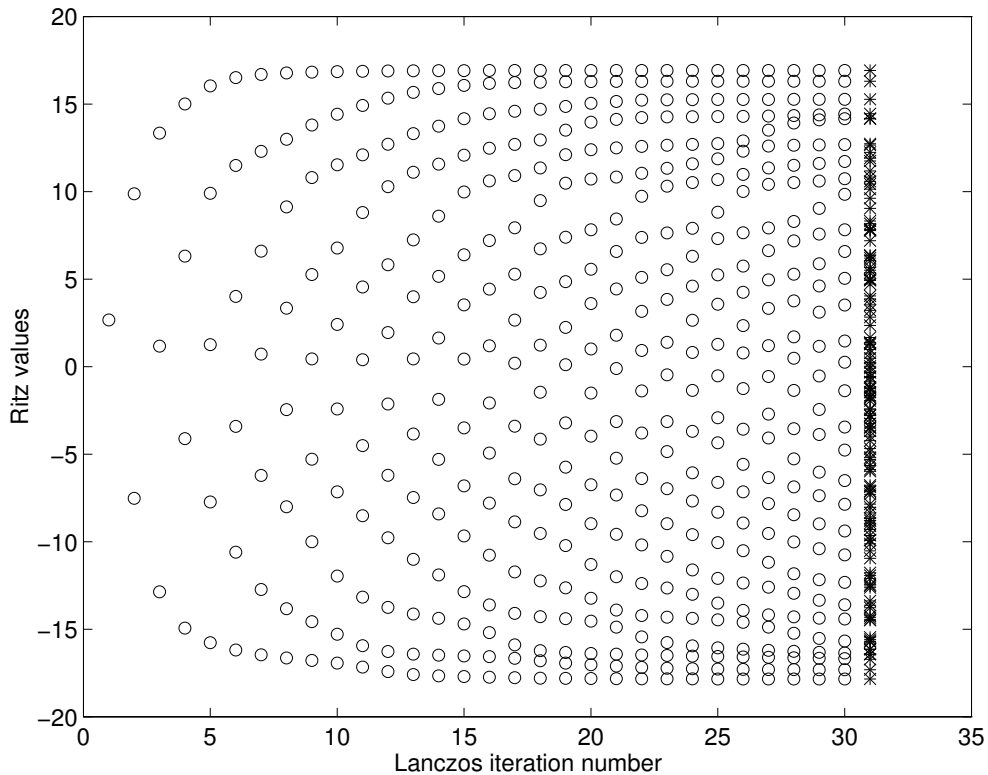
$$\begin{aligned} \|Az - \theta z\| &= \|AV_m y - \theta V_m y\| \\ &= \|(V_m H_m + f e_m^H) y - \theta V_m y\| \\ &= \|f\| |e_m^H y| \end{aligned}$$

Convergence of Lanczos/Arnoldi

- If $v_1 \in$ an m-dimensional invariant subspace of A, Arnoldi converges in m or fewer steps:

$$AV_m = V_m H_m.$$

- One rarely finds such a good v_1. In general, **extremal** and well separated eigenvalues emerge rapidly (Kaniel, Paige, Saad theory, see Parlett's book)



Computation Cost and Acceleration Methods

Cost:

- Storage for V_m, H_m: $\mathcal{O}(nm + m^2)$;
- Orthogonalization $f_m \leftarrow (I - V_m V_m^H) A v_m$: $\mathcal{O}(nm^2)$;
- Eigen-analysis of H_m: $\mathcal{O}(m^3)$;
- MATVEC $y \leftarrow Ax$: varies with applications;

Acceleration:

- Method of implicit restart $\begin{cases} \textit{polynomial} \\ \textit{rational} \end{cases}$
- Method of spectral transformation $\begin{cases} \textit{polynomial} \\ \textit{rational} \end{cases}$

Restart

Basic idea:

1. Fix the dimension m of $\mathcal{K}(A, v_1; m)$ at a moderate value;
2. Modify the starting vector

$$v_1 \leftarrow \psi(A)v_1;$$

3. Repeat the Arnoldi process with the modified v_1;

How to choose $\psi(\lambda)$? Suppose eigenvalues of A are:

$$\underbrace{\lambda_1, ..., \lambda_k,}_{\textit{wanted}} \quad \underbrace{\lambda_{k+1}, ..., \lambda_n,}_{\textit{unwanted}}$$

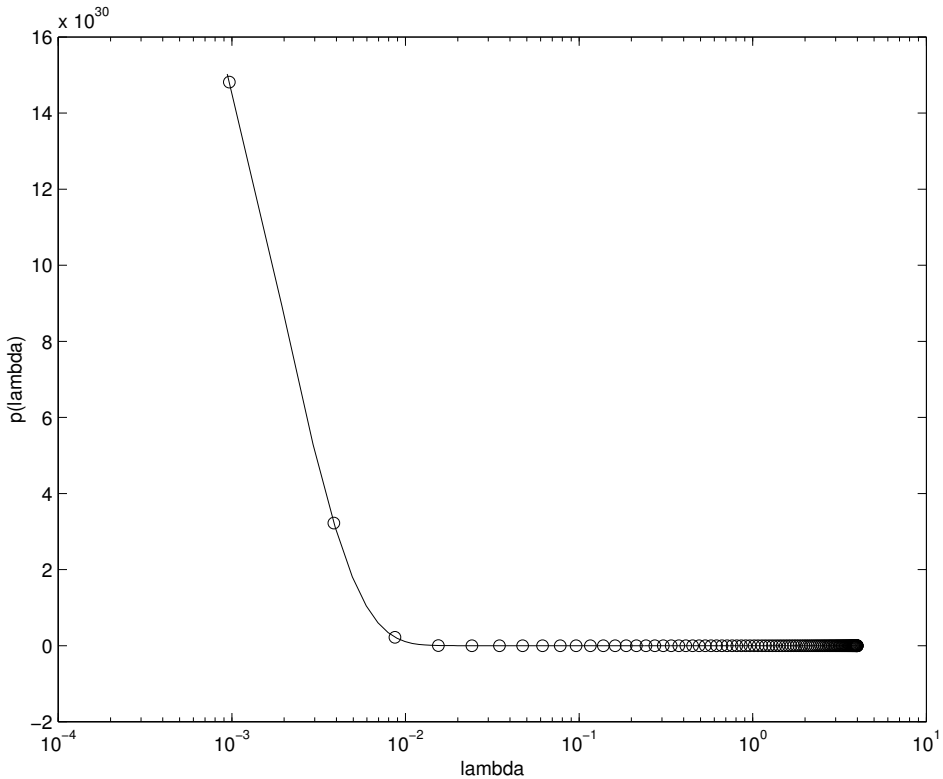
and the corresponding eigenvector are $x_1, x_2, ..., x_n$.

$$\begin{aligned} \psi(A)v_1 &= \underbrace{\gamma_1\psi(\lambda_1)x_1 + \cdots + \gamma_k\psi(\lambda_k)x_k}_{\textit{wanted}} \\ &+ \underbrace{\gamma_{k+1}\psi(\lambda_{k+1})x_{k+1} + \cdots + \gamma_n\psi(\lambda_n)x_n}_{\textit{unwanted}} \end{aligned}$$

Two types of restarts

1. $\psi(\lambda)$ is a polynomial;
2. $\psi(\lambda)$ is a rational function;

$\psi(\lambda)$ must be large on $\lambda_1, ..., \lambda_k$ and small on $\lambda_{k+1}, ..., \lambda_n$. (a polynomial filter)



Implicit Restart

1. Do not form $v_1 \leftarrow \psi(A)v_1$ explicitly;
2. Do not repeat the Arnoldi iteration from the first column;

Need to understand the connection between Arnoldi and QR (RQ)

...

QR and RQ iteration

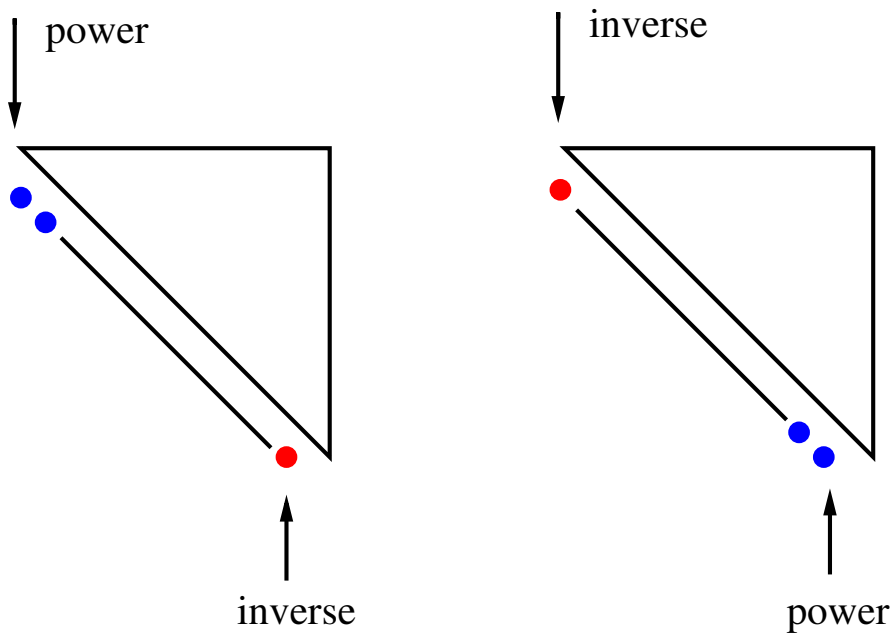
$$AV = VH$$

$$\begin{array}{c|c} \swarrow & \searrow \\ AV = VQR & AV = VRQ \\ \Downarrow & \Downarrow \\ A(VQ) = (VQ)RQ & AVQ^H = VQ^H(QR) \\ \searrow & \swarrow \end{array}$$

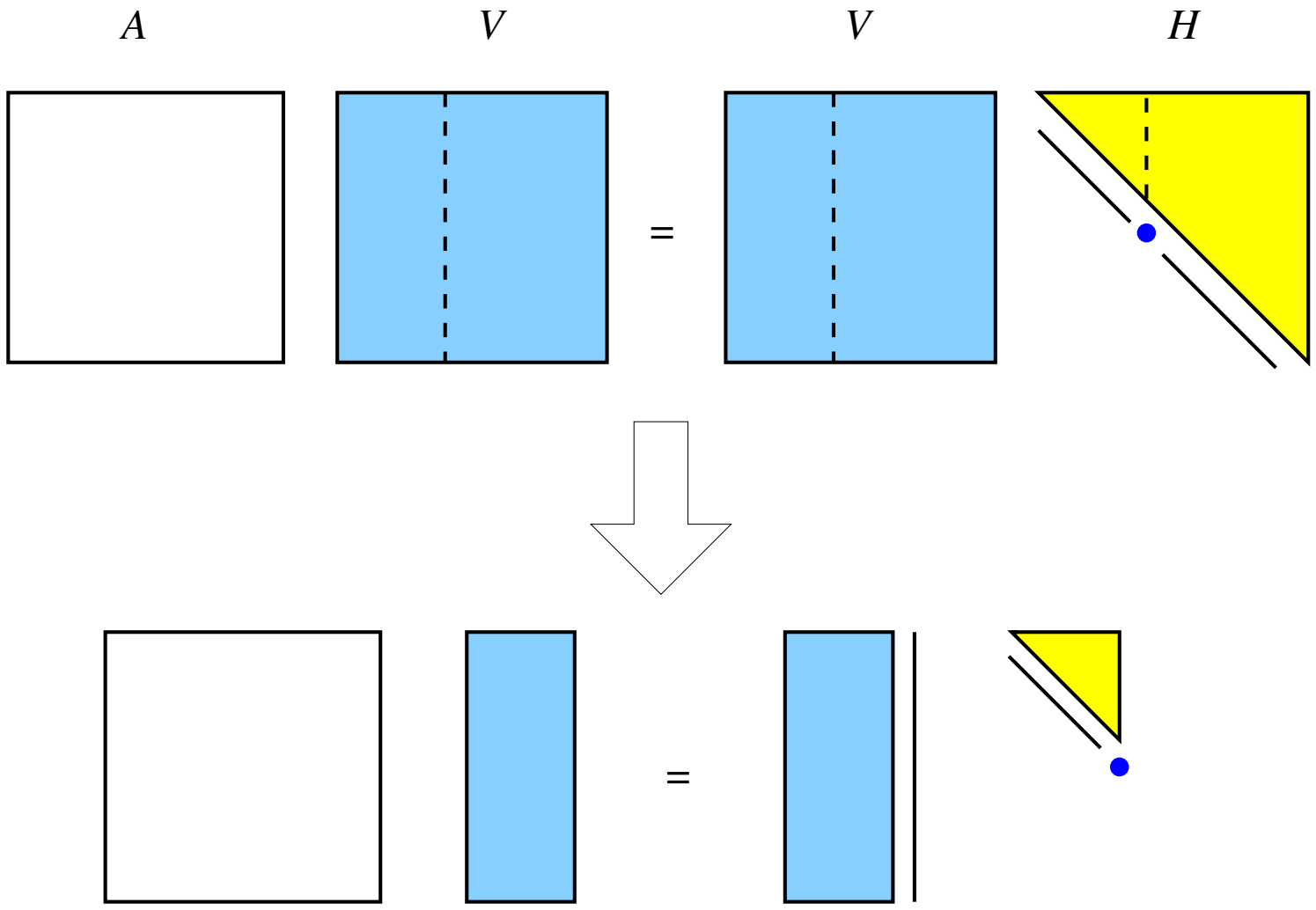
$$AV^+ = V^+H^+$$

Difference

QR	**RQ**
$AV = (VQ)R$	$A(VQ^H) = VR$
$\Downarrow$	$\Downarrow$
$Av_1 = v_1^+ \rho_{1,1}$	$Av_1^+ = v_1 \rho_{1,1}$

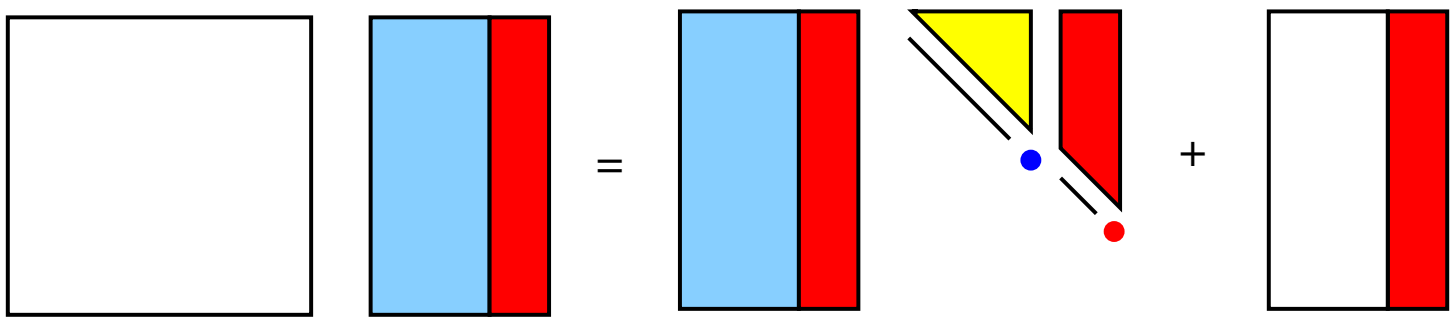


Arnoldi = Truncated Hessenberg Reduction



Implicit Researt Arnoldi = Truncated QR Iteration

$$\begin{aligned} AV_m &= V_m H_m + f e_m^H \\ AV_m &= V_m QR + f e_m^H \\ AV_m Q &= (V_m Q)(RQ) + f e_m^H Q \end{aligned}$$



$Av_1 = v_1^+ \rho_{11}$ still holds

Shifts & Polynomial filter

- Truncated Hessenberg reduction is shift-invariant

$$(A - \mu I)V_m = V_m(H_m - \mu I) + fe_m^H$$

- Applying p shifts = Running p implicitly shifted QR iterations on H_m (*bulge-chasing* algorithm)

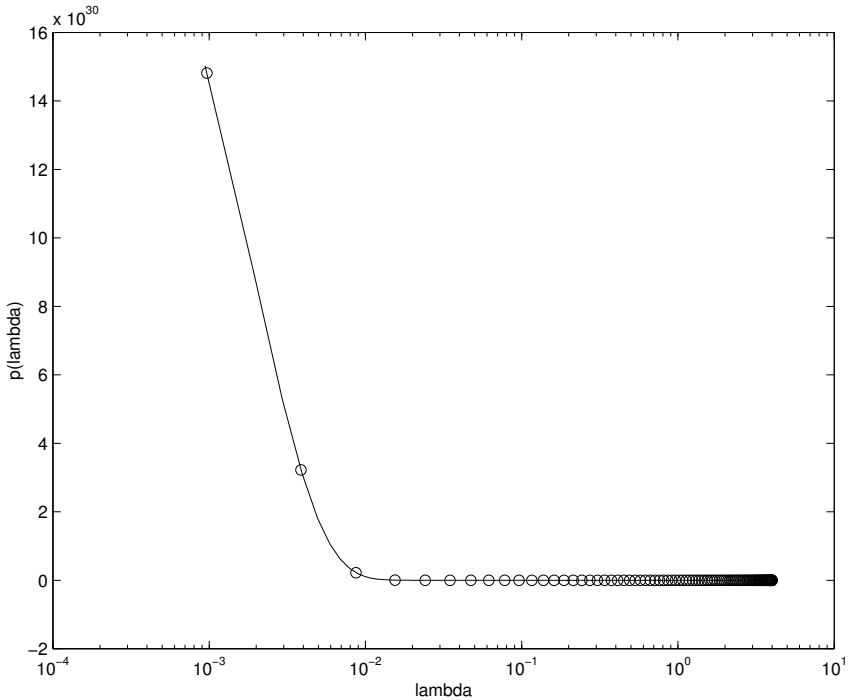
$$v_1^+ = \beta(A - \mu_1 I)(A - \mu_2 I)\cdots(A - \mu_p I)v_1$$

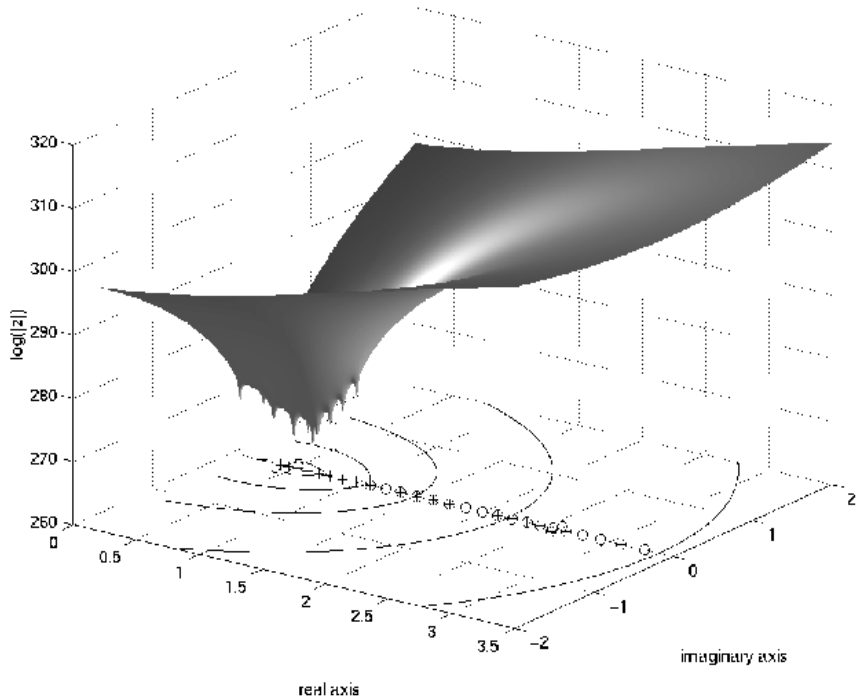
- What to use for shifts? Eigenvalues of H_m.

$$\underbrace{\theta_1, ..., \theta_k,}_{\textit{wanted}} \quad \underbrace{\theta_{k+1}, ..., \theta_m,}_{\textit{unwanted}}$$

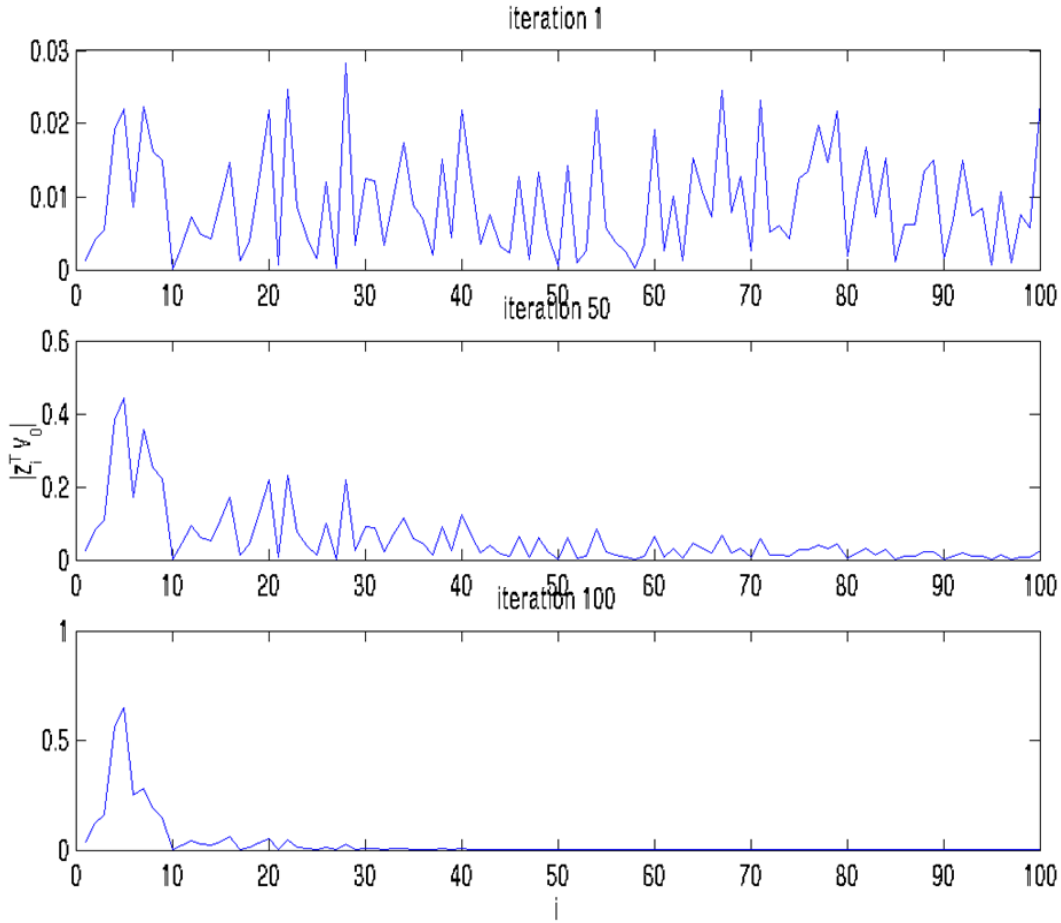
$$m = k + p$$

Filtering Polynomials





The Effect of a Polynomial Filter



The projection of v_1 onto eigenvectors of A.

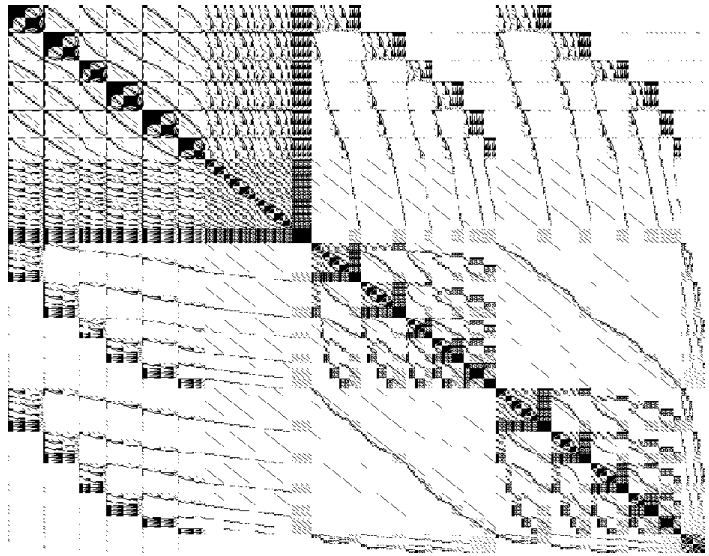
The IRA Algorithm and ARPACK

1. $AV_m = V_m H_m + f e_m^T$ (m-step Arnoldi iteration);
2. for $iter = 1, 2, \ldots$ until *convergence*
   - 2.1 Choose p shifts ($p < m$);
   - 2.2 Perform implicit QR update (bulge chase) on H_m and V_m
   - 2.3 Run p-step Arnoldi to restore truncated Hessenberg reduction;

ARPACK: http://www.caam.rice.edu/software/ARPACK

- Solve a variety of problems (sym, nonsym, real, complex)
- Location of the eigenvalues: *which* = LM, SM, LA, SA, BE, LR, SR, LI, SI
- Choose k and p (nev=k, ncv=$k + p$). p is the degree of the filtering polynomial
- Reverse communication interface
- Level-2 BLAS

Example: Nuclear CI Calculation



- size of the matrix $n = 114,735,000$
- number of eigenvalues $k = 15$
- dimension of the Krylov subspace $k + p = 30$
- number of processors used: 1,128
- time to solution: 657 sec (10 min)
- number of restarts: 21
- number of MATVEC: 558

The Limitation of a Krylov Subspace

- May require a high degree polynomial $\phi(\lambda)$ to produce an accurate approximation $z = \phi(A)v_0$;
  - Subspace of large dimension
  - Many restarts
- Spectral transformation may be prohibitively costly (sometime impossible)
- Not easy to introduce a preconditioner

The Nonlinear Equation Point of View

Formulation:

- Because $\lambda(x) = x^T Ax/x^T x$,

$$Ax = \lambda(x)x,$$

  is a nonlinear equation in x;
- Alternative formulation

$$\begin{aligned} Ax &= (x^T Ax)x \\ x^T x &= 1 \end{aligned}$$

- Many solutions;

Solve by Newton's Correction:

- Given a starting guess u such that $u^T u = 1$;
- Let $\theta = u^T A u$;
- Seek (z, δ) pair such that

$$A(u + z) = (\theta + \delta)(u + z)$$

- Ignore the 2nd order term δz (Newton correction) and impose

$$u^T z = 0$$

The Correction Equation

- Augmented form

$$\begin{pmatrix} A-\theta I & u \\ u^T & 0 \end{pmatrix} \begin{pmatrix} z \\ -\delta \end{pmatrix} = \begin{pmatrix} -r \\ 0 \end{pmatrix},$$

  where $r = Au - \theta u$;

- Projected form

$$(I - uu^T)(A - \theta I)(I - uu^T)z = -r$$

  where $u^T z = 0$

Solving the Correction Eq (Directly)

- Assume θ not converged yet, block elimination yields

$$\begin{pmatrix} I & 0 \\ u^T(A-\theta I)^{-1} & 1 \end{pmatrix} \begin{pmatrix} A-\theta I & u \\ 0 & \gamma \end{pmatrix} \begin{pmatrix} z \\ -\delta \end{pmatrix} = \begin{pmatrix} -r \\ 0 \end{pmatrix},$$

  where $\gamma = u^T(A-\theta)^{-1}u$

- $$\delta = \frac{u^T(A-\theta I)^{-1}r}{u^T(A-\theta I)^{-1}u}.$$

- Back substitution yields

$$z = \delta(A-\theta I)^{-1}u - u$$

,

Connection with the Inverse Iteration

- Adding correction z to u directly

$$x = u + z = u + \delta(A - \theta I)^{-1}u - u = \delta(A - \theta I)^{-1}u$$

- Quadratic convergence in general
- Cubic convergence for symmetric problems
- But requires solving

$$(A - \theta I)x = u$$

accurately

Jacobi-Davidson (JD)

- Solving the correction equation iteratively

$$(I - uu^T)(A - \theta I)(I - uu^T)z = -r$$

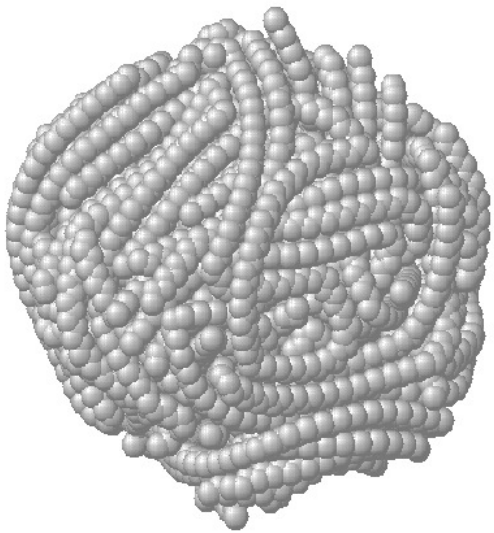
  where $u^T z = 0$
- Allows the use of a preconditioner;
- Instead of adding z to u, construct a search space $\mathcal{S} = \{u, z\}$;
- Extract Ritz pairs from $\mathcal{S}$ through Rayleigh-Ritz

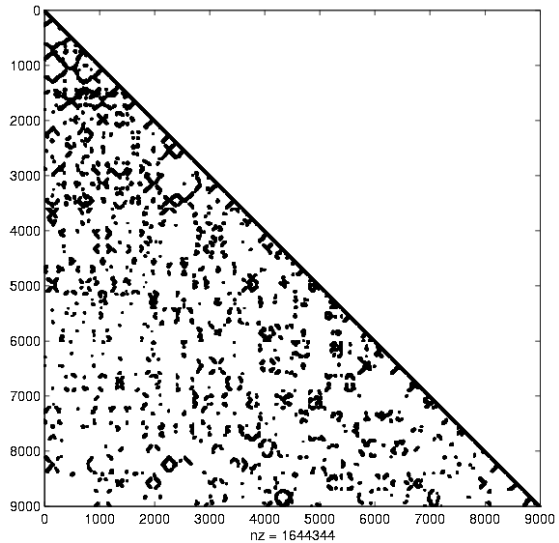
Practical issues:

- Choose an iterative solver and a preconditioner (for the correction equation);
- Set tolerance for the inner iteration;
- Shift selection;
- Restart (set a limit on the dimension of V);
- Compute more than one eigenpair (JDQR, JDQZ);

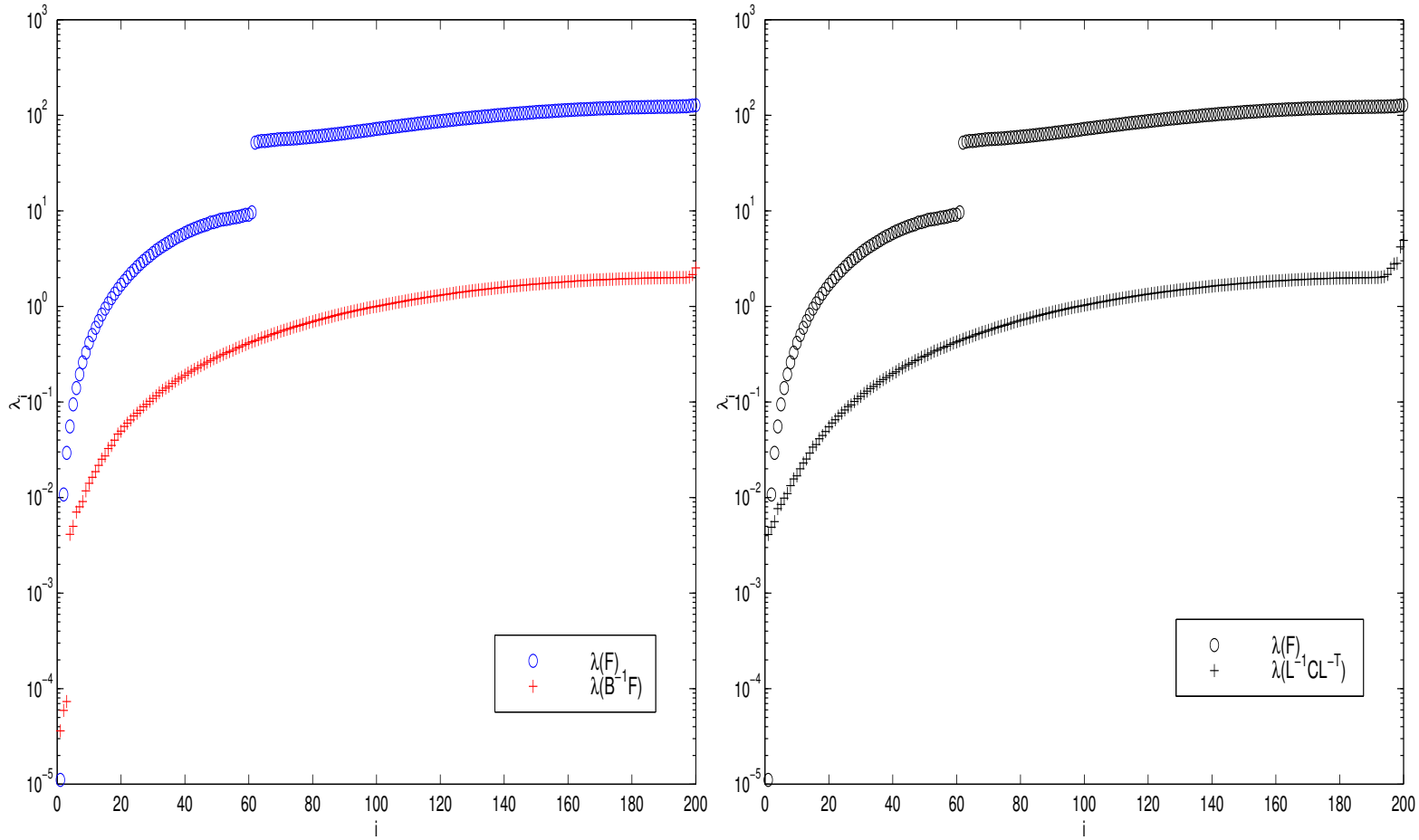
Example

- normal mode vibration analysis for macromolecules
- 3000-atom, $n = 9000$, interested in low frequency modes (small eigenvalues)

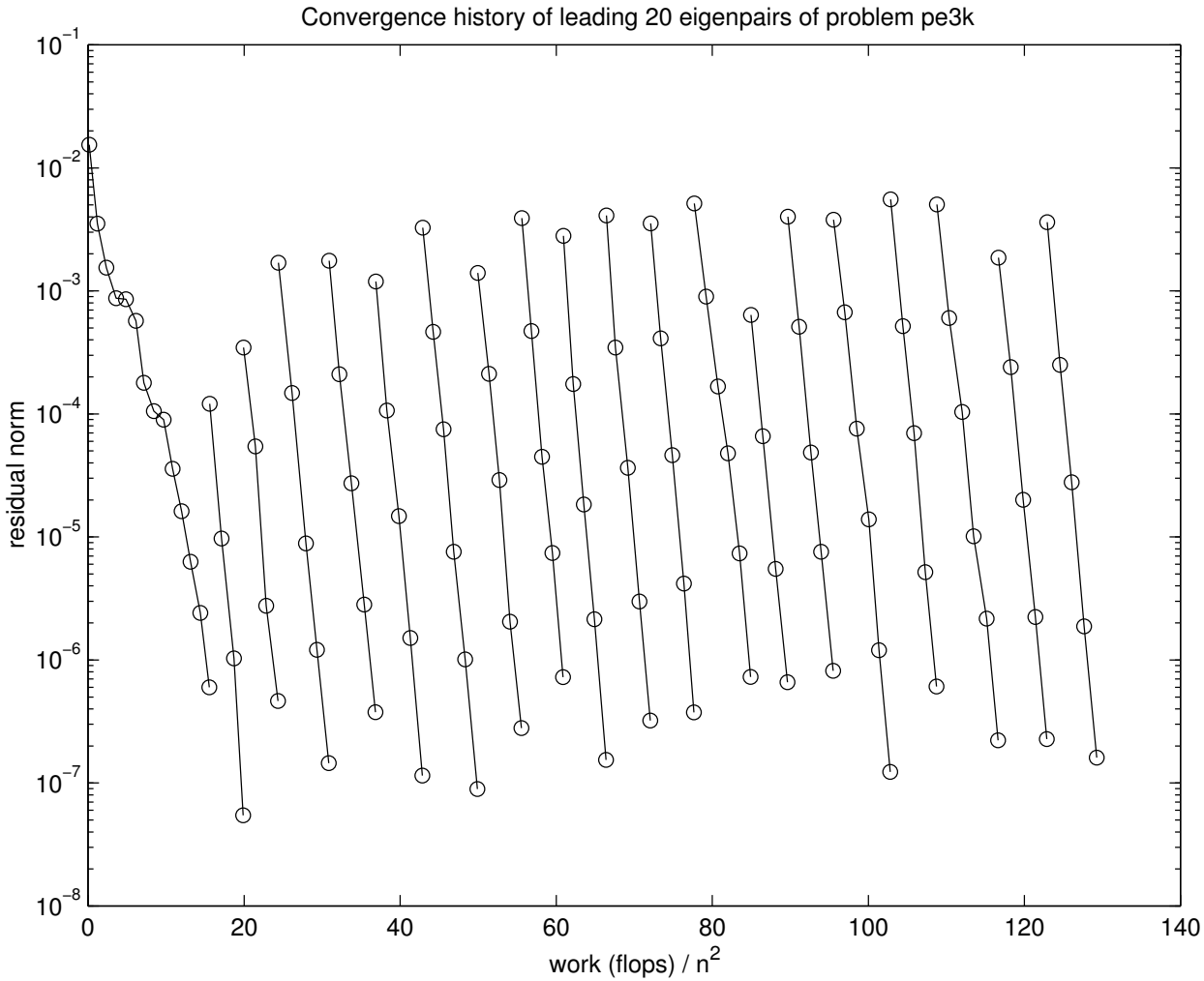




The Effect of a Preconditioner



Convergence History



The Optimization View

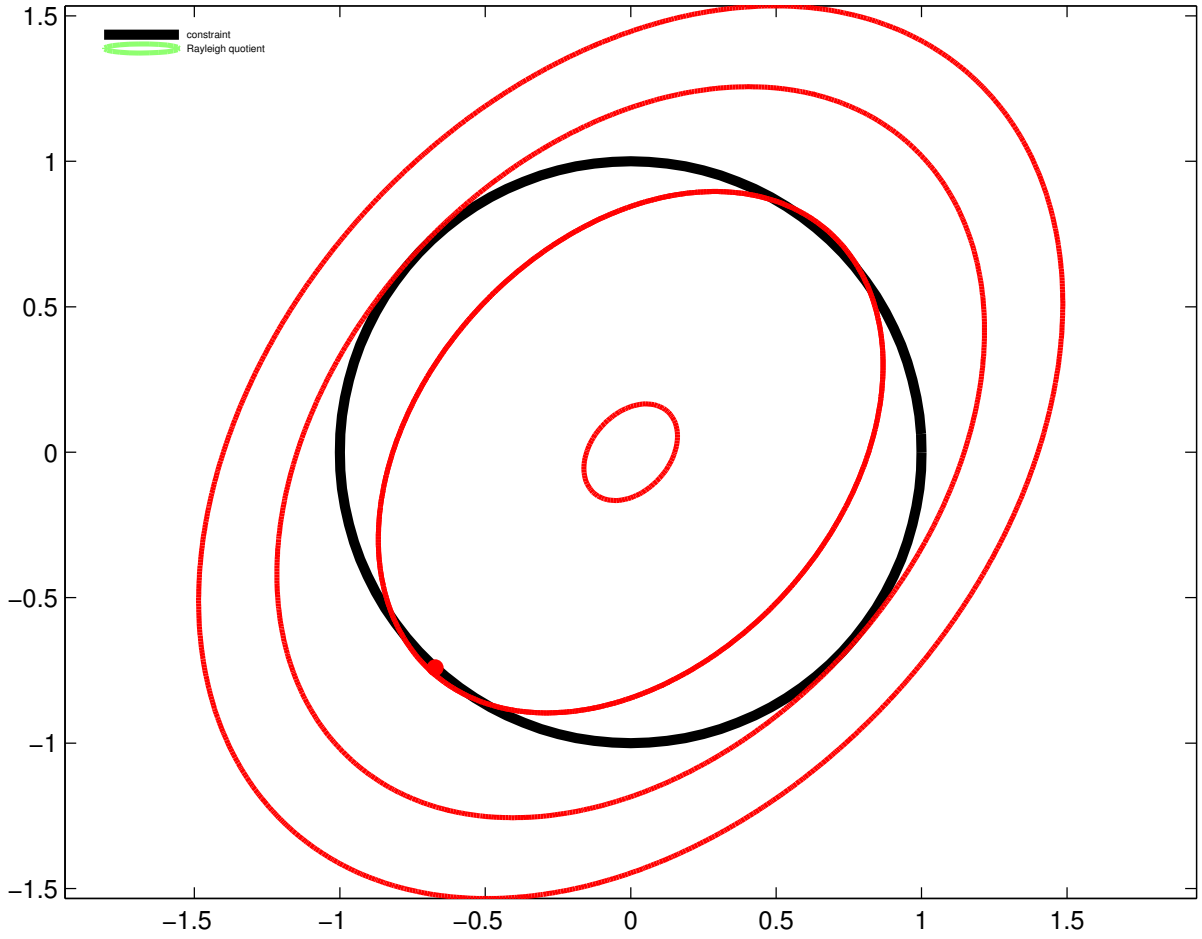
- Only valid for symmetric problems, extreme eigenvalues
- Constrained optimization

$$\min_{x^T x=1} x^T A x$$

- Lagrangian $\mathcal{L}(x, \lambda) = x^T A x - \lambda(x^T x - 1)$
- KKT condition

$$\begin{aligned} Ax - \lambda x &= 0 \\ x^T x &= 1 \end{aligned}$$

Geometry



Constrained Minimization

- Assume x_k is current approximation;
- Update by

$$x_{k+1} = \alpha x_k + \beta p_k$$

  - p_k is a descent (search) direction;
  - α, β are chosen so that
    - $x_{k+1}^T x_{k+1} = 1$;
    - $\rho(x_{k+1}) < \rho(x_k)$, where $\rho(x) = x^T A x$;

Search Direction

- Steepest descent

$$r_k = -\nabla_x \mathcal{L}(x_k, \theta_k) = -(Ax_k - \theta_k x_k)$$

- Conjugate gradient

$$p_k = p_{k-1} + \gamma r_k$$

  Choose γ so that $p_k^T A p_{k-1} = 0$

- But what about orthonormality constraint?

$$x_{k+} = \alpha x_k + \beta p_{k-1} + \gamma r_k$$

,

Subspace Minimization

- Let $V = (x_k, p_{k-1}, r_k)$, then $x_{k+1} = Vy_k$, for some y_k;
- Must solve

$$\min_{y_k^T V^T V y_k = 1} y_k^T V^T A V y_k$$

- Equivalent to solving

$$\begin{aligned} G y_k &= \lambda B y_k \\ y_k^T B y_k &= 1 \end{aligned}$$

where $B = V^T V$ and $G = V^T A V$;

Compute More Eigenpairs

- Trace minimization

$$\min_{X^T X = I_m} \frac{1}{2}\text{trace}(X^T A X)$$

  where $X \in \mathbb{R}^{n \times m}$;

- Gradient

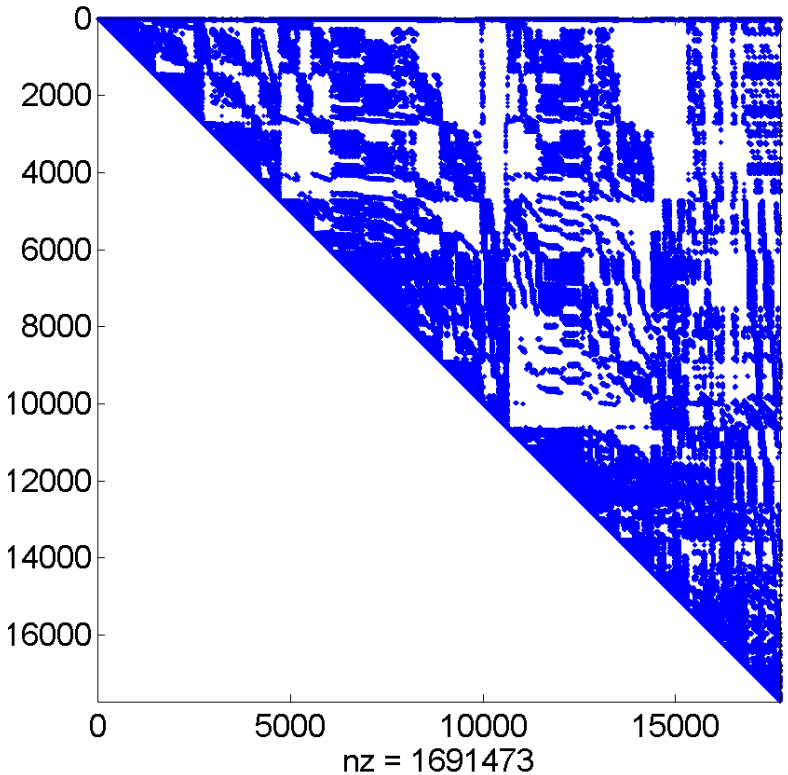
$$R_k = \nabla_x \mathcal{L}(X_k, \Lambda_k) = AX_k - X_k\Lambda_k,$$

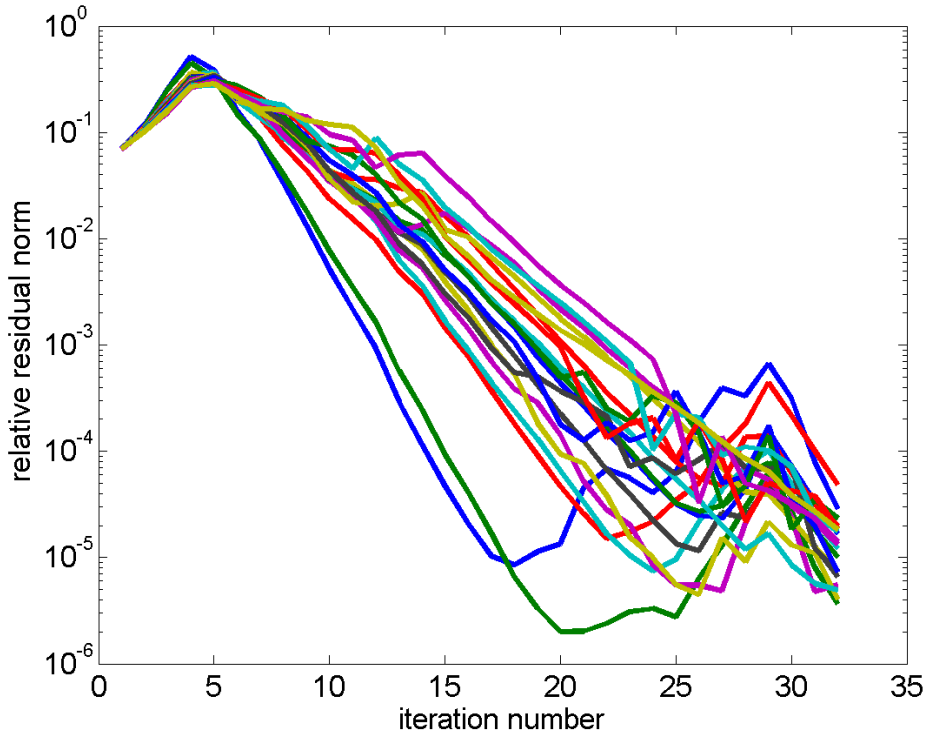
  where $\Lambda_k = X_k^T A X_k$;

Practical issues:

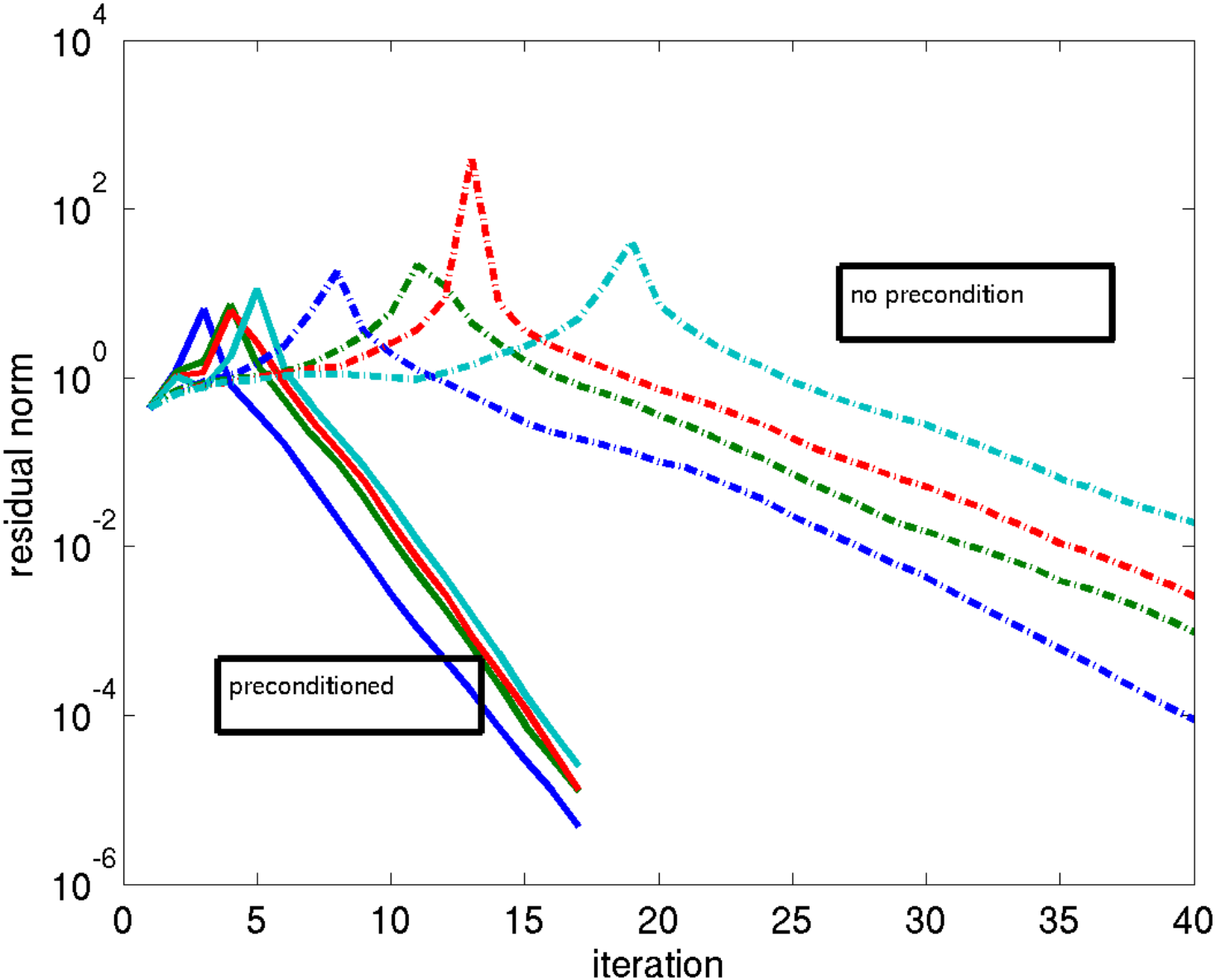
- Choice of preconditioner
- Linear dependency between columns of V;
- Deflation (not all eigenpairs converge at the same rate)
- Extension to (symmetric) generalized eigenvalue problem (straightforward)

Example: Nuclear CI calculation





Example: Electronic Structure Calculation (SiH4)



Conclusion

- It all starts from power iteration
- Krylov subspace methods
- Implcitly restarted Arnoldi and its connection with QR
- Jacobi-Davidson
- Optimization based approach