

Numerical solution of matrix eigenvalue problems

Part 2: Krylov subspace methods

Daniel Kressner
ETH Zürich

kressner@math.ethz.ch
<http://www.math.ethz.ch/~kressner>

ZSS 2008

Outline

- ▶ Krylov subspaces
- ▶ Arnoldi method
- ▶ convergence
- ▶ filtering/restarts
- ▶ shift-and-invert Arnoldi
- ▶ Lanczos method

Krylov subspaces

Krylov subspace method = Power method with memory

Krylov subspace for A and u_1 :

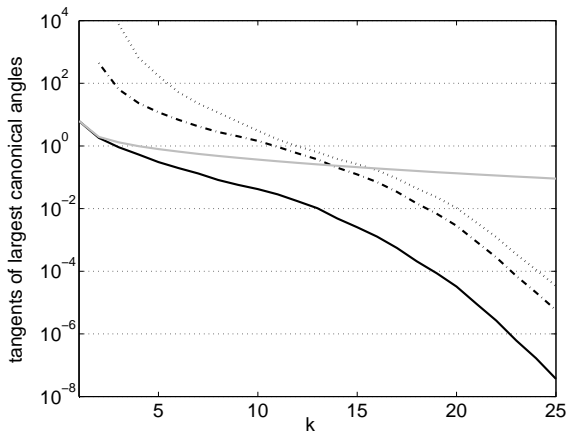
$$\mathcal{K}_k \equiv \mathcal{K}_k(A, u_1) = \text{span}\{u_1, Au_1, \dots, A^{k-1}u_1\}.$$

- ▶ Unless u_1 is contained in invariant subspace of dimension $< k$:
 $\dim(\mathcal{K}_k) = k$.
- ▶ $\mathcal{K}_k = \{p(A)u_1 : p \in \Pi_{k-1}\}$, with $\Pi_{k-1} =$ polynomials of degree at most $k - 1$.

Krylov subspaces

\mathcal{K}_k always contains better approximations to dominant eigenvector(s) than $A^{k-1}u_1$ alone.

Example: $A = \text{diag}(0.95, 0.95^2, 0.95^3, \dots)$



Krylov subspaces

Basis

$$[u_1, Au_1, \dots, A^{k-1}u_1]$$

extremely **ill-conditioned** for $k \gg 1$, useless for any practical purpose.
Construct ONB of \mathcal{K}_{k-1} by applying **Gram-Schmidt** with a twist.

Assume $U_j = [u_1, \dots, u_j]$ forms ONB of \mathcal{K}_j . Then

$$\mathcal{K}_{j+1} = \text{span}\{u_1, \dots, u_j, Au_j\},$$

\rightsquigarrow use Au_j instead of numerically dubious $A^j u_1$ in Gram-Schmidt:

$$\tilde{u}_{j+1} = (I - U_j U_j^H) Au_j, \quad u_{j+1} = \frac{1}{\|\tilde{u}_{j+1}\|_2} \tilde{u}_{j+1}.$$

Arnoldi process

Input: **start vector** u_1 with $\|u_1\|_2 = 1$, **matrix** A . Integer k .

Output: **ONB** $U_{k+1} = [u_1, \dots, u_k, u_{k+1}]$ of $\mathcal{K}_{k+1}(A, u_1)$.

for $j = 1, 2, \dots, k$ **do**

$$z_j = Au_j$$

$$h_j = U_j^H z_j$$

$$\tilde{u}_{j+1} = z_j - U_j h_j$$

$$h_{j+1,j} = \|\tilde{u}_{j+1}\|_2$$

if $h_{j+1,j} \neq 0$ **then**

break down

end if

$$u_{j+1} = \tilde{u}_{j+1} / h_{j+1,j}$$

end for

- ▶ Breakdown $\Leftrightarrow \mathcal{K}_j = \text{span}(U_j)$ is invariant subspace.
- ▶ One matrix-vector mult. / loop.
- ▶ Storage + compt. cost / loop grow proportionally with k .

Arnoldi decomposition

Arnoldi process \rightsquigarrow Arnoldi decomposition

$$AU_k = U_k H_k + h_{k+1,k} u_{k+1} e_k^T = U_{k+1} \hat{H}_k$$

with

$$H_k = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1k} \\ h_{21} & h_{22} & \cdots & h_{2k} \\ & \ddots & \ddots & \vdots \\ & & h_{k,k-1} & h_{kk} \end{bmatrix}, \quad \hat{H}_k = \begin{bmatrix} H_k \\ \hline 0 & h_{k+1,k} \end{bmatrix}.$$

If $h_{k+1,k} = 0 \rightsquigarrow \mathcal{K}_k = \text{span}(U_k)$ invariant subspace.

Ritz value/vector extraction

Galerkin condition (see subspace iteration)

$$Av - \lambda v \perp \mathcal{K}_k, \quad v \in \mathcal{K}_k.$$

Since U_k is ONB of \mathcal{K}_k , Galerkin \Leftrightarrow

$$v = U_k w, \quad U_k^T A U_k w - \mu w = 0.$$

$\rightsquigarrow (\mu, w)$ is eigenpair of $U_k^T A U_k$.

Arnoldi decomposition $\rightsquigarrow H_k = U_k^T A U_k$:

(μ, v) is **Ritz pair** if $\mu \in \Lambda(H_k)$ and $v = U_k w$ with w corresponding eigenvector of H_k .

Residual of (μ, v) has the form

$$r = Av - \lambda v = h_{k+1,k} u_{k+1} e_k^T w.$$

Thus,

$$\|r\|_2 = |h_{k+1,k}| |e_k^T w|.$$

Loss of orthogonality

Arnoldi inherits numerical instability of Gram-Schmidt. [Show loss.m](#)

Cure: Reorthogonalization.

Input: **start vector** u_1 with $\|u\|_2 = 1$, **matrix** A . Integer k .

Output: **ONB** $U_{k+1} = [u_1, \dots, u_k, u_{k+1}]$ of $\mathcal{K}_{k+1}(A, u_1)$.

for $j = 1, 2, \dots$ **do**

$$z_j = Au_j$$

$$h_j = U_j^H z_j$$

$$\tilde{u}_{j+1} = z_j - U_j h_j$$

if $\|\tilde{u}_{j+1}\|_2 < 0.7\|z_j\|_2$ **then**

$$\hat{h}_j = U_j^H \tilde{u}_{j+1}, \quad h_j \leftarrow h_j + \hat{h}_j, \quad \tilde{u}_{j+1} \leftarrow \tilde{u}_{j+1} - U_j \hat{h}_j$$

end if

$$h_{j+1,j} = \|\tilde{u}_{j+1}\|_2$$

if $h_{j+1,j} \neq 0$ **then**

break down

end if

$$u_{j+1} = \tilde{u}_{j+1} / h_{j+1,j}$$

end for

Convergence analysis

Arnoldi favors **outer parts** of the spectrum. Show arngo.m

To analyse convergence towards eigenpair (λ, x) let $X = [x, X_2]$ be invertible:

$$X^{-1}AX = \begin{bmatrix} \lambda & 0 \\ 0 & A_{22} \end{bmatrix}$$

Transform u_1 analogously: $X^{-1}u_1 = \begin{bmatrix} u_\lambda \\ u_r \end{bmatrix}$

$$\tan \theta(x, \mathcal{K}_k) \leq \kappa(X) \min_{p \in \Pi_{k-1}} \frac{\|p(A_{22})u_r\|_2}{|p(\lambda)| |u_\lambda|}.$$

\rightsquigarrow Find polynomial that is large at λ and small at all other eigenvalues.

Convergence analysis: Real spectrum

Assume A has real eigenvalues

$$\lambda_1 < \lambda_2 \leq \dots \leq \lambda_n$$

Idea of proof:

- ▶ Set p to $(k - 1)$ th Chebyshev polynomial on $[\lambda_2, \lambda_n]$.
- ▶ Then $|p(\lambda_j)| \leq 1$ for $j > 1$ and

$$p(\lambda_1) \geq \frac{1}{2}\rho^{k-1}, \quad \rho = 1 + 2\frac{\lambda_2 - \lambda_1}{\lambda_n - \lambda_2} > 1.$$

$$\tan \theta(x_1, \mathcal{K}_k) \leq C\rho^{-k} \tan(x_1, u_1).$$

Polynomial filtering

Idea: Replace start vector u_1 by $q(A)u_1$, where q is large at the wanted eigenvalues and small in the region of unwanted eigenvalues.

Information on the region of unwanted eigenvalues is obtained iteratively, during the Arnoldi process.

1. Perform m steps of Arnoldi. Cluster Ritz values into **wanted** μ_1, \dots, μ_k and **unwanted** μ_{k+1}, \dots, μ_m .
2. Choose $q(A)$ such that $\deg(q) < m$ and q is **small** at unwanted but **large** at wanted part of the spectrum.
3. Replace $u_1 \leftarrow q(A)u_1 / \|q(A)u_1\|_2$ and go to step 1.

Remarks:

1. **No extra** matrix-vector multiplies in Step 3.
2. **Lossy** compression of information contained in \mathcal{K}_m into a single vector.
3. If zeros of q are Ritz values \Rightarrow filtering with **exact shifts**.

Thick/implicit restarts

Goal: Maintain complete information contained in **wanted** Ritz vectors.

Input: Arnoldi decomposition

$$AU_m = U_m H_m + h_{m+1,m} u_{m+1} e_m^T$$

with **wanted** μ_1, \dots, μ_k and **unwanted** μ_{k+1}, \dots, μ_m Ritz values.

Compute **ordered Schur decomposition** (see `schord` in MATLAB):

$$Q^T H_m Q = T = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}, \quad \begin{aligned} \Lambda(T_{11}) &= \{\mu_1, \dots, \mu_k\}, \\ \Lambda(T_{22}) &= \{\mu_{k+1}, \dots, \mu_m\}. \end{aligned}$$

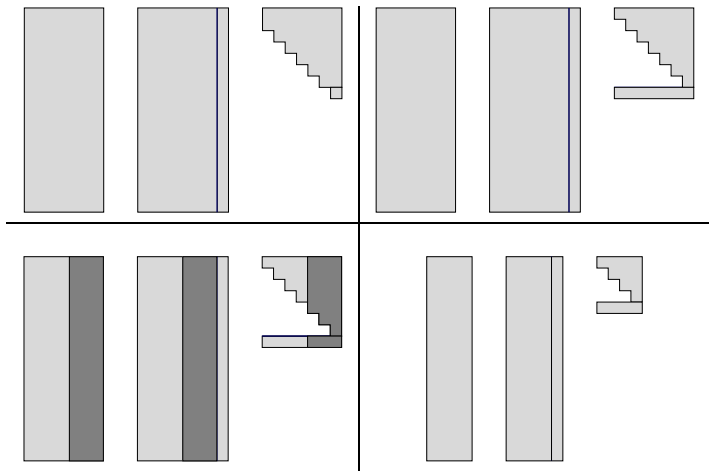
Transform

$$A \hat{U}_m = \hat{U}_m T + u_{m+1} \hat{b}_m^T, \quad \hat{U}_m = U_m Q, \quad b_m = h_{m+1,m} Q^T e_m,$$

and truncate

$$A \hat{U}_k = \hat{U}_k T_{11} + \hat{u}_{k+1} \hat{b}_k^T.$$

Thick/implicit restarts



Thick/implicit restarts

- ▶ Implicit restarting mathematically equivalent to explicit restarting with $\hat{u}_1 = q(A)u_1 / \|q(A)u_1\|_2$ and performing k steps of Arnoldi, where $q(z) = (z - \mu_{k+1}) \cdots (z - \mu_m)$.
- ▶ After restart, truncated Arnoldi-like decomposition (also called Krylov-Schur decomposition) is expanded again to order m by applying Arnoldi process. Whole procedure is repeated until all wanted eigenvalues have converged.
- ▶ Typical choice: $m = 2k$.
- ▶ Converged Ritz vectors are locked by reordering to top left corner of T .
- ▶ If A is symmetric, T is diagonal \rightsquigarrow reordering reduces to permutation.

Shift-and-invert Arnoldi

Numerical experiments/convergence analysis suggest: **convergence to interior eigenvalues is poor**. To speed up convergence: Replace A by $(A - \tau I)^{-1}$ with **target** $\tau \in \mathbb{C}$. show arngo2.m

Not suited for inexact application of $(A - \tau I)^{-1}$!

Variants:

- ▶ If eigenvalues close to **several** τ_1, \dots, τ_m are required.

Rational Arnoldi:

Arnoldi decomposition for $(A - \tau_j I)^{-1}$

\implies Arnoldi decomposition for $(A - \tau_{j+1} I)^{-1}$

Allows to efficiently transfer information gained from one target to another.

- ▶ **Polynomial preconditioning:** Construct fixed low-degree polynomial p supposedly small on unwanted eigenvalues. Replace A by $p(A)$.

Lanczos process

Let A symmetric. H_k produced by Arnoldi inherits symmetry and is thus tridiagonal. In *theory*, Lanczos is a special case of Arnoldi for symmetric matrices.

Input: **start vector** u_1 with $\|u\|_2 = 1$, **symmetric matrix** A . Integer k .
Output: **ONB** $U_{k+1} = [u_1, \dots, u_k, u_{k+1}]$ of $\mathcal{K}_{k+1}(A, u_1)$.

```
for  $j = 1, 2, \dots$  do  
   $z_j = Au_j$   
   $\alpha_j = u_j^T z_j$   
   $\tilde{u}_{j+1} = z_j - u_j \alpha_j$   
  if  $j > 1$  then  
     $\tilde{u}_{j+1} \leftarrow \tilde{u}_{j+1} - u_{j-1} \beta_{j-1}$   
  end if  
   $\beta_j = \|\tilde{u}_{j+1}\|_2$   
   $u_{j+1} = \tilde{u}_{j+1} / \beta_j$   
end for
```

- ▶ Only 2–3 vectors of length n need to be stored to define the iteration.

Lanczos process

Lanczos decomposition:

$$AU_k = U_k H_k + \beta_k u_{k+1} e_k^T, \quad H_k = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \beta_{k-1} & \alpha_k \end{bmatrix}.$$

Loss of orthogonality: Run loss.m and losslan.m

- ▶ happens much faster than with Arnoldi;
- ▶ appearance of **ghost eigenvalues**;
- ▶ strongly related to convergence of Ritz pair (Paige);
- ▶ convergence only partially affected; eigenvalue-based convergence bounds remain valid (Paige).

Reorthogonalization

Full reorthogonalization is an effective but **expensive cure**:show nolosslan.m

- ▶ requires to store *complete* U_k ;
- ▶ compt. cost grows from $O(mvp + nk)$ to $O(mvp + nk^2)$.

Paige's theorem Consider a k -order Lanczos decomposition computed in floating point arithmetic with macheps ε . The Ritz pairs $(\mu_1, v_1), \dots, (\mu_k, v_k)$ satisfy

$$v_i^T q_{k+1} = \frac{O(\varepsilon \|A\|_2)}{\|r_i\|_2}, \quad i = 1, \dots, k,$$

with $r_i = Av_i - \mu_i v_i$.

Paige's theorem suggests **selective reorthogonalization** only vs. converged Ritz pairs (within $\sqrt{\varepsilon}$). More advanced strategies, e.g., by [Simon'84].

Not mentioned...

- ▶ Block Lanczos/Arnoldi.
- ▶ Nonsymmetric Lanczos.
- ▶ Lanczos for SVD.
- ▶ Purification strategies.
- ▶ ...

Literature

- ▶ The MATLAB command `eigs` is based on ARPACK, a comprehensive implementation of the Arnoldi method with implicit restarts. Some recent developments are not (yet) included in this package. This particularly concerns the command `svds` for computing partial SVDs. Other software can be found under <http://www.netlib.org/utk/people/JackDongarra/la-sw.html>.
- ▶ A more detailed discussion of convergence results for Arnoldi can be found in [Y. Saad. Numerical methods for large eigenvalue problems. Online available from Saad's web page.]
- ▶ The way implicit restarting is described here can be found in [G. W. Stewart. Matrix Algorithms. Vol. II. 2001]. ARPACK uses a different algorithm that is mathematically but *not* numerically equivalent.

Literature

- ▶ The Lanczos process and finite precision aspects are discussed in detail in [B. N. Parlett. *The Symmetric Eigenvalue Problem*. SIAM, 1997] and [G erard Meurant. *The Lanczos and Conjugate Gradient Algorithms: From Theory to Finite Precision Computations*. SIAM, 2006].
- ▶ Many practical issues concerning the implementation of Lanczos and Arnoldi are detailed in [J. K. Cullum and R. A. Willoughby. *Lanczos Algorithms for Large Symmetric Eigenvalue Computations*. Birkh user, 1985] and [R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK Users' Guide*. SIAM, 1998.], respectively.
- ▶ Efficient Krylov subspace methods for computing partial SVDs can be found in [Baglama and Reichel. *Augmented Implicitly Restarted Lanczos Bidiagonalization Methods* *SIAM J. Sci. Comput.*, 27 (2005), pp. 19–42.].