Solving large scale eigenvalue problems

Lecture 10, May 4, 2016: More on Lanczos and Arnoldi

http://people.inf.ethz.ch/arbenz/ewp/

Peter Arbenz
Computer Science Department, ETH Zürich
E-mail: arbenz@inf.ethz.ch
Survey of today’s lecture

▶ Restarting Arnoldi
▶ Thick restarts for Lanczos
▶ Krylov-Schur algorithm
▶ Rational Krylov space methods
▶ Polynomial filtering
Reminder: the Arnoldi algorithm

- The Arnoldi algorithm constructs orthonormal bases for the Krylov spaces

\[ \mathcal{K}^j(x) = \mathcal{K}^j(x, A) := \mathcal{R}([x, Ax, \ldots, A^{j-1}x]) \subseteq \mathbb{R}^{n \times j}, \quad j = 1, 2, \ldots \]

- Let \( \{v_1, \ldots, v_j\} \) be an orthonormal bases for \( \mathcal{K}^j(x, A) \).
  We obtain \( v_{j+1} \) by orthogonalizing \( Av_j \) against \( \{v_1, \ldots, v_j\} \):

\[
r_j = Av_j - V_j V_j^* Av_j = Av_j - \sum_{i=1}^{j} v_i (v_i^* A v_j),
\]

\[
v_{j+1} = r_j / \| r_j \|.
\]

- The Lanczos algorithm is symmetric version of Arnoldi.
Restarting Arnoldi and Lanczos algorithms

- The number of iteration steps can be very high in Arnoldi/Lanczos algorithms.
- Iteration count depends on properties of the matrix (distribution of its eigenvalues) but also on initial vectors.
- High iteration counts entail a large memory requirement and a high amount of computation (reorthogonalization).
- **Implicitely restarted Arnoldi/Lanczos algorithms** reduce these costs by limiting the dimension of the search space [1].
- Iteration is stopped after a number of steps, dimension of search space is reduced, and finally the Arnoldi / Lanczos iteration is resumed.
- **IRA** is implemented in ARPACK and in the matrix eigensolver *eigs* in MATLAB.
**m-step Arnoldi iteration with implicit restart**

After execution of \( m \)-step Arnoldi algorithm: Arnoldi relation

\[
AV_m = V_m H_m + r_m e_m^*, \quad H_m = \begin{bmatrix} \vdots \\ \end{bmatrix}
\]

with \( r_m = \beta_m v_{m+1}, \quad \|v_{m+1}\| = 1. \)

We apply \( k < m \) implicit QR steps with shifts \( \mu_1, \ldots, \mu_k \) to \( H_m \):

1. \( H_m^+ := H_m. \)
2. \textbf{for} \( i := 1, \ldots, k \) \textbf{do}
3. \( H_m^+ := V_i^* H_m^+ V_i, \) \text{ with } \( H_m^+ - \mu_i I = V_i R_i \) (QR factorization)
4. \textbf{end for}
$m$-step Arnoldi iteration with implicit restart (cont.)

Let

\[ V^+ := V_1 V_2 \cdots V_k \quad V^+ = \begin{bmatrix} V_1 & V_2 & \cdots & V_k \end{bmatrix} \]

\[ V_{m}^+ := V_m V^+, \quad H_{m}^+ := (V^+)^* H_m V^+. \]

Then,

\[ AV_{m}^+ = V_{m}^+ H_{m}^+ + r_m e_m^* V^+. \quad (1) \]

Since

\[ e_m^* V^+ = (0, \ldots, 0, *, \ldots, *), \quad k + p = m. \]

\[ p - 1 \quad k + 1 \]

we discard last $k$ columns in (1) to obtain an Arnoldi relation of length $p$.

Executing $k$ iterations steps we re-establish an Arnoldi relation of length $m$. 
Choice of shifts

- In ARPACK [1] and elsewhere, all eigenvalues of $H_m$ are computed. Those $k$ eigenvalues that are furthest away from some target value are chosen as shifts.
- Since the new Arnoldi relation has the initial vector

  $$
  v_1^{(k)} \leftarrow \psi(A)v_1, \quad \psi(\lambda) = \prod_{i=1}^{k} (\lambda - \mu_i)
  $$

  the components in the direction of undesired eigenvalues is ‘deemphasized’.
Convergence

Let $H_m s = s \vartheta$, $\|s\| = 1$, $\hat{x} = V_m s$ the associated Ritz vector. Then,

$$\|A \hat{x} - \vartheta \hat{x}\| = \|AV_m s - V_m H_m s\| = \|r_m\| |e_m^* s| = \beta_m |e_m^* s|.$$ 

A Hermitian: Theorem of Krylov–Bogoliubov provides interval that contains eigenvalue of $A$.

In ARPACK, a Ritz pair $(\vartheta, \hat{x})$ is considered converged if

$$\beta_m |e_m^* s| \leq \max(\varepsilon M \|H_m\|, tol \cdot |\vartheta|).$$

As $|\vartheta| \leq \|H_m\| \leq \|A\|$, the inequality $\|E\| \leq tol \cdot \|A\|$ holds at convergence. Well-conditioned eigenvalues are well approximated.
Discussion of IRA / IRL

- Implicit restarting procedures: very clever ways to get rid of unwanted directions in the search space and still keeping a Lanczos or Arnoldi basis.
- The latter admits to continue the iteration in a known framework.
- Lanczos / Arnoldi relations admit very efficient checks for convergence.
- The restart has the effect of altering the starting vector.
- Now we discuss algorithms that work with Krylov spaces but are not restricted to Arnoldi / Lanczos bases.
When is a basis generating a Krylov space?

Let \( \mathbf{v}_1, \ldots, \mathbf{v}_k \) be linearly independent \( n \)-vectors. Is the subspace \( \mathcal{V} := \text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_k\} \) a Krylov space, i.e., is there a vector \( \mathbf{q} \in \mathcal{V} \) such that \( \mathcal{V} = \mathcal{K}_k(A, \mathbf{q}) \)?

**Theorem**

\( \mathcal{V} = \text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_k\} \) is a Krylov space if and only if there is a \( k \)-by-\( k \) matrix \( M \) such that

\[
R := AV - VM, \quad V = [\mathbf{v}_1, \ldots, \mathbf{v}_k],
\]

has rank one and \( \text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_k, R(R)\} \) has dimension \( k + 1 \).

For a proof see the lecture notes or paper by Stewart [2].
The Lanczos algorithm with thick restarts

Apply theorem to the case where a subspace is spanned by some Ritz vectors. For $A = A^*$ a Lanczos relation is given by

$$AQ_k - Q_k T_k = \beta_{k+1} q_{k+1} e_k^T.$$  \hspace{1cm} (3)

Let spectral decomposition of tridiagonal $T_k$ be

$$T_k S_k = S_k \Theta_k, \quad S_k = [s_1^{(k)}, \ldots, s_k^{(k)}], \quad \Theta_k = \text{diag}(\vartheta_1, \ldots, \vartheta_k).$$

Then, for all $i$, the Ritz vector

$$y_i = Q_k s_i^{(k)} \in \mathcal{K}_k(A, q)$$

gives rise to the residual

$$r_i = Ay_i - y_i \vartheta_i = \beta_{k+1} q_{k+1} s_{ki}^{(k)} \in \mathcal{K}_{k+1}(A, q) \ominus \mathcal{K}_k(A, q).$$
The Lanczos algorithm with thick restarts (cont.)

So, for any set of indices $1 \leq i_1 < \cdots < i_j \leq k$ we have

$$A[y_{i_1}, y_{i_2}, \ldots, y_{i_j}] - [y_{i_1}, y_{i_2}, \ldots, y_{i_j}] \text{diag}(\vartheta_{i_1}, \ldots, \vartheta_{i_j}) = \beta_{k+1} q_{k+1} [s^{(k)}_{k,i_1}, s^{(k)}_{k,i_2}, \ldots, s^{(k)}_{k,i_j}].$$

Theorem $\implies$ any set $[y_{i_1}, y_{i_2}, \ldots, y_{i_j}]$ of Ritz vectors forms a Krylov space.

The generating vector differs for each set \{i_1, \cdots, i_j\}.

Split the indices 1, \ldots, k in two sets.

- First set: ‘good’ Ritz vectors that we want to keep and that we collect in $Y_1$
- Second set: ‘bad’ Ritz vectors that we want to remove. They go into $Y_2$. 

The Lanczos algorithm with thick restarts (cont.)

\[
A[Y_1, Y_2] - [Y_1, Y_2] \begin{bmatrix} \Theta_1 \\ \Theta_2 \end{bmatrix} = \beta_{k+1} q_{k+1} [s_1^*, s_2^*].
\] (4)

Keeping the first set of Ritz vectors and purging (deflating) the rest yields

\[AY_1 - Y_1 \Theta_1 = \beta_{k+1} q_{k+1} s_1^*.\]

We now can restart a Lanczos procedure by orthogonalizing \(Aq_{k+1}\) against \(Y_1 =: [y_1^*, \ldots, y_j^*]\) and \(q_{k+1}\).
The Lanczos algorithm with thick restarts (cont.)

From the equation

$$A\mathbf{y}_i - \mathbf{y}_i \varphi_i = \mathbf{q}_{k+1} \sigma_i, \quad \sigma_i = \beta_{k+1} \mathbf{e}_k^* \mathbf{s}_i^{(k)}$$

we get

$$\mathbf{q}^*_{k+1} A \mathbf{y}_i = \sigma_i,$$

whence

$$\mathbf{r}_{k+1} = A \mathbf{q}_{k+1} - \beta_k \mathbf{q}_{k+1} - \sum_{i=1}^{j} \sigma_i \mathbf{y}_i \perp \mathcal{K}_{k+1}(A, \mathbf{q}).$$

From this point on the Lanczos algorithm proceeds with the ordinary three-term recurrence.
The Lanczos algorithm with thick restarts (cont.)

We finally arrive at relation $AQ_m - Q_m T_m = \beta_{m+1} q_{m+1} e_m^T$ with

$$Q_m = [y_1, \ldots, y_j, q_{k+1}, \ldots, q_{m+k-j}]$$

and

$$T_m = \begin{pmatrix}
\vartheta_1 & \sigma_1 & & \\
\vdots & \ddots & \ddots & \\
\vartheta_j & \sigma_j & & \\
\sigma_1 & \cdots & \sigma_j & \alpha_{k+1} & \\
& \ddots & \ddots & \ddots & \\
& & \ddots & \ddots & \ddots \\
& & & \beta_{m+k-j-1} & \\
& & & \beta_{m+k-j-1} & \alpha_{m+k-j}
\end{pmatrix}$$
The Lanczos algorithm with thick restarts (cont.)

- This procedure, called **thick restart**, has been suggested by Wu & Simon [3].
- It allows to restart with any number of Ritz vectors.
- In contrast to the implicitly restarted Lanczos procedure, we need the spectral decomposition of $T_m$. (Its computation is not an essential overhead.)
- The spectral decomposition admits a simple sorting of Ritz values.
- We could further split the first set of Ritz pairs into converged and unconverged ones, depending on the value $\beta_{m+1} |s_k,i|$. If this quantity is below a given threshold we set the value to zero and lock (deflate) the corresponding Ritz vector, i.e., accept it as an eigenvector.
Thick restart Lanczos algorithm

1: Let us be given \( k \) Ritz vectors \( y_i \) and a residual vector \( r_k \) such that \( A y_i = \vartheta_i y_i + \sigma_i r_{k+1}, \ i = 1, \ldots, k \). The value \( k \) may be zero in which case \( r_0 \) is the initial guess. This algorithm computes an orthonormal basis \( y_1, \ldots, y_k, q_{k+1}, \ldots, q_m \) that spans a \( m \)-dimensional Krylov space whose generating vector is not known unless \( k = 0 \).

2: \( z := A q_{k+1} \);
3: \( \alpha_{k+1} := q_{k+1}^* z \);
4: \( r_{k+1} = z - \alpha_{k+1} q_{k+1} - \sum_{i=1}^{k} \sigma_i y_i \);
5: \( \beta_{k+1} := \| r_{k+1} \| \);
6: for \( i = k + 2, \ldots, m \) do
7: \( q_i := r_{i-1} / \beta_{i-1} \);
8: \( z := A q_i \);
Thick restart Lanczos algorithm (cont.)

9: \[ \alpha_i := q_i^*z; \]
10: \[ r_i = z - \alpha_i q_i - \beta_{i-1} q_{i-1} \]
11: \[ \beta_i = \|r_i\| \]
12: end for

- Problem of losing orthogonality is similar to plain Lanczos.
- In their numerical experiments the simplest procedure, full reorthogonalization, performs similarly or even faster than the more sophisticated reorthogonalization procedures.
Krylov–Schur algorithm

Krylov–Schur algorithm (Stewart [2]) is a generalization of the thick-restart procedure for non-Hermitian problems.

The Arnoldi algorithm constructs the Arnoldi relation

\[ AQ_m = Q_m H_m + r_m e_m^*, \]

where \( H_m \) is Hessenberg and \([Q_m, r_m]\) has full rank. Let \( H_m = S_m T_m S_m^* \) be a Schur decomposition of \( H_m \) with unitary \( S_m \) and upper triangular \( T_m \). Similarly as before we have

\[ AY_m = Y_m T_m + r_m s^*, \quad Y_m = Q_m S_m, \quad s^* = e_m^* S_m. \quad (5) \]
Krylov–Schur algorithm (cont.)

The upper trangular form of $T_m$ eases analysis of Ritz pairs. It admits moving unwanted Ritz values to the lower-right of $T_m$. We collect ‘good’ and ‘bad’ Ritz vectors in matrices $Y_1$ and $Y_2$:

$$A[Y_1, Y_2] - [Y_1, Y_2] \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} = \beta_{k+1} q_{k+1} [s^*_1, s^*_2]. \quad (6)$$

Keeping the first set of Ritz vectors and purging the rest yields

$$AY_1 - Y_1 T_{11} = \beta_{k+1} q_{k+1} s^*_1.$$
Krylov–Schur algorithm (cont.)

Thick-restart Lanczos: eigenpair considered found if $\beta_{k+1}|s_{ik}|$ sufficiently small.
Determination of converged subspace not so easy with general Krylov–Schur.
However, if we manage to bring $s_1$ into the form

$$s_1 = \begin{bmatrix} s'_1 \\ s''_1 \end{bmatrix} = \begin{bmatrix} 0 \\ s''_1 \end{bmatrix}$$

then we found an invariant subspace:

$$A[Y'_1, Y''_1] - [Y'_1, Y''_1] \begin{bmatrix} T'_{11} & T'_{12} \\ T'_{21} & T'_{22} \end{bmatrix} = \beta_{k+1}q_{k+1}[0^T, s''_1^*]$$

i.e.,

$$AY'_1 = Y'_1 T'_{11}$$
Krylov–Schur algorithm (cont.)

- In most cases $s'_1$ consists of a single small element or of two small elements in the case of a complex-conjugate eigenpair of a real nonsymmetric matrix [2].
- These small elements are then declared zero and the columns in $Y'_1$ are locked, i.e., they are not altered anymore in the future computations.
- Orthogonality against them has to be enforced in the continuation of the eigenvalue computation though.
The rational Krylov space method

- Assume that we have computed a number of Ritz pairs in the neighborhood of some shift $\sigma_1$ with SI-Lanczos/Arnoldi algorithms of generalized eigenvalue problem

$$Ax = \lambda Bx.$$ 

- Can we restart with a changed shift $\sigma_2$?
- $\implies$ rational Krylov space method
The rational Krylov space method (cont.)

We start out with Arnoldi relation

\[(A - \sigma_1 B)^{-1}BQ_j = Q_j H_j + r_j e^T = Q_{j+1} \tilde{H}_j\]  \hspace{1cm} (7)

or, using the Schur decomposition of \(H_j\), \(H_j = S_j T_j S_j^*\),

\[(A - \sigma_1 B)^{-1}B Y_j = Y_j T_j + r_j s^* = Y_{j+1} \begin{bmatrix} T_j \\ s^* \end{bmatrix}, \hspace{1cm} Y_{j+1} = [Y_j, r_j] \]  \hspace{1cm} (8)

We want to derive a Krylov–Schur relation for a new shift \(\sigma_2 \neq \sigma_1\) from (8) for the same space \(\mathcal{R}(Y_{j+1})\) without accessing the matrices \(A\) or \(B\).
The rational Krylov space method (cont.)

Want to replace basis $Y_{j+1}$ by a new basis $W_{j+1}$, that spans the same subspace as $Y_{j+1}$ but can be interpreted as the orthonormal basis of a Krylov–Schur relation with the new shift $\sigma_2$.

We rewrite the relation (8) as

$$BY_j = BY_{j+1} \begin{bmatrix} I_j & 0^* \end{bmatrix} = (A - \sigma_1 B) Y_{j+1} \begin{bmatrix} T_j \\ s^* \end{bmatrix}.$$ 

Introducing the shift $\sigma_2$ this becomes

$$BY_{j+1} \left\{ \begin{bmatrix} I_j \\ 0^* \end{bmatrix} + (\sigma_1 - \sigma_2) \begin{bmatrix} T_j \\ s^* \end{bmatrix} \right\} = (A - \sigma_2 B) Y_{j+1} \begin{bmatrix} T_j \\ s^* \end{bmatrix}.$$ 

To construct a Krylov–Schur relation we must get rid of the last non-zero row of the matrix in braces.
The rational Krylov space method (cont.)

Use the QR factorization

\[
\begin{bmatrix}
I_j \\
0_T
\end{bmatrix} + (\sigma_1 - \sigma_2) \begin{bmatrix}
T_j \\
S^*
\end{bmatrix} = Q_{j+1} \begin{bmatrix}
R_j \\
0_T
\end{bmatrix}.
\]

Then,

\[
BY_{j+1}Q_{j+1} \begin{bmatrix}
R_j \\
0_T
\end{bmatrix} \equiv BW_{j+1} \begin{bmatrix}
R_j \\
0_T
\end{bmatrix}
= BW_j R_j = (A - \sigma_2 B) W_{j+1} Q_{j+1}^* \begin{bmatrix}
T_j \\
S^*
\end{bmatrix}
\]
The rational Krylov space method (cont.)

Multiplying with \((A - \sigma_2 B)^{-1}\) from the left we obtain

\[
(A - \sigma_2 B)^{-1} BW_j = W_{j+1} Q_{j+1}^* \begin{bmatrix} T_j R_j^{-1} \\ s^* \end{bmatrix} = W_{j+1} \begin{bmatrix} M_j \\ t^* \end{bmatrix}
\]

or

\[
(A - \sigma_2 B)^{-1} BW_j = W_j M_j + w_{j+1} t^*. 
\]

This equation can easily been transformed into an Arnoldi or Lanczos relation.
The rational Krylov space method (cont.)

- In a practical implementation, the mentioned procedure is combined with locking, purging, and implicit restart.
- First run SI Arnoldi with first shift $\sigma_1$. When eigenvalues around $\sigma_1$ have converged, lock the good Ritz vectors and purge the bad ones, leaving an Arnoldi (7).
- Then introduce new shift $\sigma_2$ and perform the steps above to get a new basis $W_{j+1}$ that replaces $V_{j+1}$. Start at new shift by operating on the last vector of new basis

$$r := (A - \sigma_2 B)^{-1} B w_{j+1}$$

and get next basis vector $w_{j+2}$ in the new Arnoldi recurrence.
- Continue until eigenpairs around $\sigma_2$ have converged.
- Repeat the same procedure with further shifts.
Polynomial filtering

- SI-Lanczos/Arnoldi are the standard way to compute eigenvalues close to some target/shift $\sigma$.
- This works as long as the factorization of $A - \sigma I$ (or of $A - \sigma B$) is feasible.
- An alternative that has emerged in the last few years is polynomial filtering: Replace the matrix $A$ by $p(A)$ where $p \in \mathbb{P}_d$ for some $d$.

$$Au_i = \lambda_i u_i \implies p(A)u_i = p(\lambda_i)u_i.$$ 

Eigenvectors remain (at least if $\lambda_i \neq \lambda_j \implies p(\lambda_i) \neq p(\lambda_j)$).
Polynomial filtering (cont.)

- Let's assume that $A = A^*$, whence $\sigma(A) \in \mathbb{R}$.
- Let's further assume that $\sigma(A) \in [-1, 1]$. (This requires some knowledge about $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$). **Scaling.**
- If we wanted to compute eigenvalues in the interval $[.3, .6]$ we could choose $p(\lambda) = 1 - 0.4 (\lambda - 0.45)^2$:

![Graph showing polynomial filtering](image)

- Eigenvalues are poorly separated $\rightarrow$ slow convergence.
Polynomial filtering (cont.)

- The spectral decomposition of $A \in \mathbb{F}^{n \times n}$ is given by

$$A = \sum_{i=1}^{n} \lambda_i u_i u_i^*.$$  

(Multiple eigenvalues are taken into account according to their multiplicities.) Spectral projector for interval $[a, b]$: 

$$P_{[a,b]} = \sum_{a \leq \lambda_i \leq b} u_i u_i^*.$$  

(It may be useful to know how many eigenvalues $\lambda_i$ are in $[a, b]$.) 

$$P_{[a,b]} = \sum_{i=1}^{n} \chi_{[a,b]}(\lambda_i) u_i u_i^* = \chi_{[a,b]}(A), \quad \chi_{[a,b]}(x) = \begin{cases} 1, & x \in [a, b], \\ 0, & \text{otherwise}. \end{cases}$$
Polynomial filtering (cont.)

- To actually apply $\chi_{[a,b]}(A)$ we would need to know the eigenvectors $u_i$ associated with the eigenvalues $\lambda_i \in [a, b]$.
- An idea is to approximate $\chi_{[a,b]}$ by a polynomial. We could write
  $$\chi_{[a,b]}(x) = \sum_{j=0}^{d} \gamma_j T_j(x),$$
  where $T_j(x), j = 0, 1, \ldots$, are Chebyshev polynomials.
- Chebyshev polynomials $T_j(x) = \cos(j \arccos x) \equiv \cos(j \vartheta)$ are orthogonal with respect to the inner product
  $$\langle f, g \rangle \equiv \int_{-1}^{1} \frac{f(x)g(x)}{\sqrt{1-x^2}} \, dx = \int_{0}^{\pi} f(\cos \vartheta)g(\cos \vartheta) \, d\vartheta.$$
  Note: $x = \cos \vartheta \quad \longrightarrow \quad dx = -\sin \vartheta \, d\vartheta$. 

Large scale eigenvalue problems, Lecture 10, May 4, 2016
Polynomial filtering (cont.)

- Any function $f$ with $\langle f, f \rangle < \infty$ can be expanded in a series of Chebyshev polynomials:

$$f(x) = \sum_{j=0}^{\infty} \frac{\langle f, T_j \rangle}{\langle T_j, T_j \rangle} T_j(x).$$

- For $f = \chi_{[a,b]}$ we get

$$\gamma_j = \frac{\langle \chi_{[a,b]}, T_j \rangle}{\langle T_j, T_j \rangle} = \begin{cases} \frac{1}{\pi} (\arccos(a) - \arccos(b)), & j = 0, \\ \frac{2}{\pi} \sin(\arccos(a)) - \sin(\arccos(b)) \cdot \frac{1}{j}, & j > 0. \end{cases}$$

By truncation we get a polynomial approximation $p \in \mathbb{P}_d$ of $\chi_{[a,b]}$ that is optimal in the norm $\langle \cdot, \cdot \rangle^{1/2}$. 
Polynomial filtering (cont.)

- The relation

\[
\cos(k + 1) \vartheta + \cos(k - 1) \vartheta = 2 \cos \vartheta \cos k \vartheta
\]

gives rise to the three-term recurrence

\[
T_{k+1}(x) = 2x T_k(x) - T_{k-1}(x), \quad k > 0, \quad T_0(x) = 1, \quad T_1(x) = x.
\]

- Let \( t_k = T_k(A)x \). Then \( t_0 = T_0(A)x = Ix \) and \( t_1 = T_1(A)x = Ax \), and

\[
t_{k+1} = 2A t_k - t_{k-1}, \quad k > 0.
\]

- Since the \( p(x) \approx \chi_{[a,b]}(x) \) oscillate very much at the discontinuities of \( \chi_{[a,b]} \) (Gibbs phenomenon), the coefficients \( \gamma_k \) are often damped. Jackson damping is popular, see e.g. Schofield, Chelikowsky, Saad: *A spectrum slicing method for the Kohn–Sham problem*. Computer Phys. Comm. **183** (2012) 487–505.
Polynomial filtering (cont.)

Filter of degree $d = 40$ for $[a, b] = [.3, .6]$. 
Polynomial filtering (cont.)

Filter of degree $d = 80$ for $[a, b] = [.3, .6]$. 
Polynomial filtering (cont.)

- Very large eigenvalue problems have huge memory requirements
- A solution: spectrum slicing:
  - Divide spectrum in ‘slices’ / ‘windows’ of a few hundred or thousand eigenvalues at a time.
  - Deceivingly simple looking idea for computing interior eigenvalues, generating parallelism.
- Issues:
  - Deal with interfaces: duplicate/missing eigenvalues
  - Window size: need estimate number of eigenvalues
  - Polynomial degree
References


