Solving large scale eigenvalue problems

Lecture 10, May 2, 2018: 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

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

$$R := AV - VM, \qquad V = [\mathbf{v}_1, \ldots, \mathbf{v}_k], \tag{1}$$

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

For a proof see the lecture notes or paper by Stewart [1].

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

$$AQ_k - Q_k T_k = \beta_{k+1} \mathbf{q}_{k+1} \mathbf{e}_k^T. \tag{2}$$

Let spectral decomposition of tridiagonal T_k be

$$T_k S_k = S_k \Theta_k, \qquad S_k = [\mathbf{s}_1^{(k)}, \ldots, \mathbf{s}_k^{(k)}], \quad \Theta_k = \mathrm{diag}(\vartheta_1, \ldots, \vartheta_k).$$

Then, for all i, the Ritz vector

$$\mathbf{y}_i = Q_k \mathbf{s}_i^{(k)} \in \mathcal{K}_k(A, \mathbf{q})$$

gives rise to the residual

$$\mathbf{r}_i = A\mathbf{y}_i - \mathbf{y}_i \vartheta_i = \beta_{k+1} \mathbf{q}_{k+1} s_{ki}^{(k)} \in \mathcal{K}_{k+1}(A, \mathbf{q}) \ominus \mathcal{K}_k(A, \mathbf{q}).$$

The Lanczos algorithm with thick restarts (cont.)

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

$$A[\mathbf{y}_{i_1}, \mathbf{y}_{i_2}, \ldots, \mathbf{y}_{i_j}] - [\mathbf{y}_{i_1}, \mathbf{y}_{i_2}, \ldots, \mathbf{y}_{i_j}]\,\mathrm{diag}(\vartheta_{i_1}, \ldots, \vartheta_{i_j})$$
$$= \beta_{k+1}\mathbf{q}_{k+1}[s^{(k)}_{k,i_1}, s^{(k)}_{k,i_2}, \ldots, s^{(k)}_{k,i_j}].$$

Theorem $\implies$ any set $[\mathbf{y}_{i_1}, \mathbf{y}_{i_2}, \ldots, \mathbf{y}_{i_j}]$ 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.)

$$A[Y_1, Y_2] - [Y_1, Y_2] \begin{bmatrix} \Theta_1 & \\ & \Theta_2 \end{bmatrix} = \beta_{k+1} \mathbf{q}_{k+1} [\mathbf{s}_1^*, \mathbf{s}_2^*]. \qquad (3)$$

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

$$AY_1 - Y_1\Theta_1 = \beta_{k+1} \mathbf{q}_{k+1} \mathbf{s}_1^*.$$

We now can restart a Lanczos procedure by orthogonalizing $A\mathbf{q}_{k+1}$ against $Y_1 =: [\mathbf{y}_1^*, \ldots, \mathbf{y}_j^*]$ and $\mathbf{q}_{k+1}$.

The Lanczos algorithm with thick restarts (cont.)

From the equation

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

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 $AQ_m - Q_m T_m = \beta_{m+1} \mathbf{q}_{m+1} \mathbf{e}_m^T$ with

$$Q_m = [\mathbf{y}_1, \ldots, \mathbf{y}_j, \mathbf{q}_{k+1}, \ldots, \mathbf{q}_{m+k-j}]$$

and

$$T_m = \begin{pmatrix} \vartheta_1 & & & \sigma_1 & & \\ & \ddots & & \vdots & & \\ & & \vartheta_j & \sigma_j & & \\ \sigma_1 & \cdots & \sigma_j & \alpha_{k+1} & \ddots & \\ & & & \ddots & \ddots & \beta_{m+k-j-1} \\ & & & & \beta_{m+k-j-1} & \alpha_{m+k-j} \end{pmatrix}$$

The Lanczos algorithm with thick restarts (cont.)

- This procedure, called thick restart, has been suggested by Wu & Simon [2].
- 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 $\mathbf{y}_i$ and a residual vector $\mathbf{r}_k$ such that $A\mathbf{y}_i = \vartheta_i \mathbf{y}_i + \sigma_i \mathbf{r}_{k+1}$, $i = 1, \dots, k$. The value k may be zero in which case $\mathbf{r}_0$ is the initial guess.
This algorithm computes an orthonormal basis $\mathbf{y}_1, \dots, \mathbf{y}_k, \mathbf{q}_{k+1}, \dots, \mathbf{q}_m$ that spans a m-dimensional Krylov space whose generating vector is not known unless $k = 0$.
2: $\mathbf{z} := A\mathbf{q}_{k+1}$;
3: $\alpha_{k+1} := \mathbf{q}_{k+1}^* \mathbf{z}$;
4: $\mathbf{r}_{k+1} = \mathbf{z} - \alpha_{k+1}\mathbf{q}_{k+1} - \sum_{i=1}^{k} \sigma_i \mathbf{y}_i$
5: $\beta_{k+1} := \|\mathbf{r}_{k+1}\|$
6: **for** $i = k+2, \dots, m$ **do**
7: $\quad \mathbf{q}_i := \mathbf{r}_{i-1}/\beta_{i-1}$.
8: $\quad \mathbf{z} := A\mathbf{q}_i$;

Thick restart Lanczos algorithm (cont.)

$\quad \alpha_i := \mathbf{q}_i^* \mathbf{z};$

$\quad \mathbf{r}_i = \mathbf{z} - \alpha_i \mathbf{q}_i - \beta_{i-1} \mathbf{q}_{i-1}$

$\quad \beta_i = \|\mathbf{r}_i\|$

end for

- Problem of losing orthogonality is similar to plain Lanczos.
- Wu–Simon [2] investigate various reorthogonalizing strategies known from plain Lanczos (full, selective, partial).
- 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 [1]) is generalization of the thick-restart procedure for non-Hermitian problems.

The Arnoldi algorithm constructs the Arnoldi relation

$$AQ_m = Q_m H_m + \mathbf{r}_m \mathbf{e}_m^*,$$

where H_m is Hessenberg and $[Q_m, \mathbf{r}_m]$ 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

$$AY_m = Y_m T_m + \mathbf{r}_m \mathbf{s}^*, \qquad Y_m = Q_m S_m, \quad \mathbf{s}^* = \mathbf{e}_m^* S_m. \tag{4}$$

Krylov–Schur algorithm (cont.)

The upper triangular 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:

$$A[Y_1, Y_2] - [Y_1, Y_2] \begin{bmatrix} T_{11} & T_{12} \\ & T_{22} \end{bmatrix} = \beta_{k+1} \mathbf{q}_{k+1} [\mathbf{s}_1^*, \mathbf{s}_2^*]. \qquad (5)$$

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

$$AY_1 - Y_1 T_{11} = \beta_{k+1} \mathbf{q}_{k+1} \mathbf{s}_1^*.$$

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 $\mathbf{s}_1$ into the form

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

then we found an invariant subspace:

$$A[Y_1', Y_1''] - [Y_1', Y_1''] \begin{bmatrix} T_{11}' & T_{12}' \\ & T_{22}' \end{bmatrix} = \beta_{k+1}\mathbf{q}_{k+1}[\mathbf{0}^T, \mathbf{s}_1''^*]$$

i.e.,

$$AY_1' = Y_1' T_{11}'$$

Krylov–Schur algorithm (cont.)

- In most cases $\mathbf{s}_1'$ consists of a single *small* element or of two small elements in the case of a complex-conjugate eigenpair of a real nonsymmetric matrix [1].
- 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.

Polynomial filtering

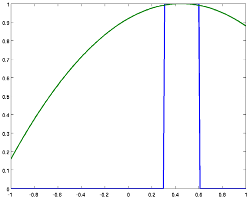
- SI-Lanczos/Arnoldi are the standard way to compute eigenvalues close to some target/shift σ.
- This works as long as the factorization of $A - \sigma I$ (or of $A - \sigma B$) is feasible.
- An alternative that has emerged in the last few years is polynomial filtering:
  Replace the matrix A by $p(A)$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

$$A\mathbf{u}_i = \lambda_i \mathbf{u}_i \quad \Longrightarrow \quad p(A)\mathbf{u}_i = p(\lambda_i)\mathbf{u}_i.$$

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



- Eigenvalues are poorly separated $\longrightarrow$ slow convergence.

Polynomial filtering (cont.)

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

$$A = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{u}_i^*.$$

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

$$P_{[a,b]} = \sum_{a\le\lambda_i\le b} \mathbf{u}_i \mathbf{u}_i^*.$$

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

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

Polynomial filtering (cont.)

- To actually apply $\chi_{[a,b]}(A)$ we would need to know the eigenvectors $\mathbf{u}_i$ 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: $x = \cos\vartheta \longrightarrow dx = -\sin\vartheta\, d\vartheta.$

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

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

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

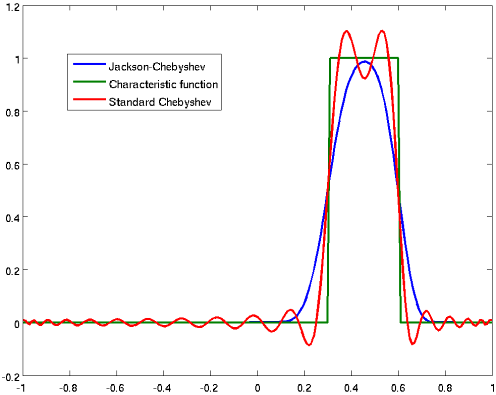
$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x),\ k>0, \quad T_0(x)=1,\, T_1(x)=x.$$

- Let $\mathbf{t}_k = T_k(A)\mathbf{x}$. Then $\mathbf{t}_0 = T_0(A)\mathbf{x} = I\mathbf{x}$ and $\mathbf{t}_1 = T_1(A)\mathbf{x} = A\mathbf{x}$, and

$$\mathbf{t}_{k+1} = 2A\mathbf{t}_k - \mathbf{t}_{k-1}, \qquad k>0.$$

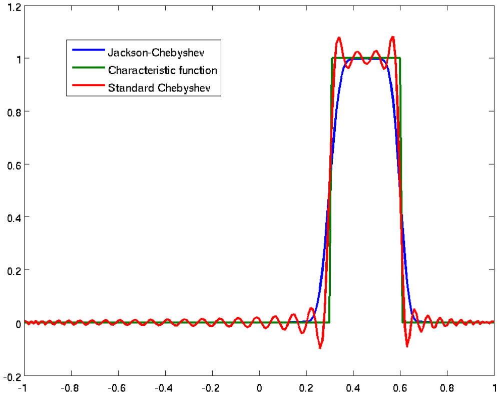
- Since the $p(x) \approx \chi_{[a,b]}(x)$ oscillate very much at the discontinuities of $\chi_{[a,b]}$ (Gibbs phenomenon), the coefficients γ_k are often *damped*. Jackson damping is popular,
see e.g. Schofield, Chelikowsky, Saad [3].

Polynomial filtering (cont.)



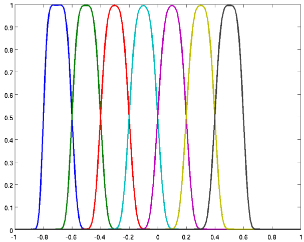
Filter of degree $d = 40$ for $[a, b] = [.3, .6]$.

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:
- Devide 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 to estimate number of eigenvalues in slice
  - Polynomial degree increases with shrinking slice width

References

[1] G. W. Stewart, *A Krylov–Schur algorithm for large eigenproblems*, SIAM J. Matrix Anal. Appl., **23** (2001), pp. 601–614.

[2] K. Wu and H. D. Simon, *Thick-restart Lanczos method for large symmetric eigenvalue problems*, SIAM J. Matrix Anal. Appl., **22** (2000), pp. 602–616.

[3] G. Schofield, J. R. Chelikowsky, Y. Saad: *A spectrum slicing method for the Kohn–Sham problem*. Comput. Phys. Comm. **183** (2012), pp. 487–505.