Utrecht, 7 december 2011

Eigenvalues and eigenvectors II

Gerard Sleijpen

Universiteit Utrecht
Department of Mathematics

http://www.staff.science.uu.nl/∼sleij101/

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

with $\mathbf{A}$ $n \times n$.

Today's topic.

- Computing approximate eigenvalues
- Expanding search subspaces.

Program

- Subspace methods
- Expansion
- Krylov subspace approach
  Arnoldi, **Shift and Invert** Arnoldi
- Convergence
- Accelerated **R**ayleigh **Q**uotient **I**teration
  Rational **K**rylov **S**equence method
- Optimal expansion
  Jacobi-Davidson
- Restart
  LOCG, **I**mplicitly **R**estarted **A**rnoldi **M**ethod
- Deflation

Subspace methods

Iterative. In each step

- **Expansion.** Expand the search subspace.
- **Extraction.** Extract an appropriate approximate solution from the search subspace.

Practical aspects.

Construct basis $\mathbf{v}_1, \ldots, \mathbf{v}_k$ of the search subspace.

Notation. $\mathbf{V}_k = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$.

- Compute expansion vector $\mathbf{t}$, "orthogonalise" to $\mathbf{v}_{k+1}$.
  orthogonalize, $\mathbf{A}$-orthogonalize, bi-orthogonalize, ...
- Approximate solution $\mathbf{x}_k = \mathbf{V}_k y_k$; compute y_k.

[Hestenes Stiefel '52]

$\mathbf{A}^* = \mathbf{A} > 0$, Conjugate Gradient

$$
\begin{array}{l}
\mathbf{x} = \mathbf{0},\ \mathbf{r} = \mathbf{b},\ \mathbf{u} = \mathbf{0},\ \rho = 1 \\
\texttt{While}\ \ \|\mathbf{r}\| > \mathit{tol}\ \ \texttt{do} \\
\quad \sigma = -\rho,\ \ \rho = \mathbf{r}^*\mathbf{r},\ \ \beta = \rho/\sigma \\
\quad \mathbf{u} \leftarrow \mathbf{r} - \beta\,\mathbf{u},\ \ \mathbf{c} = \mathbf{A}\mathbf{u} \\
\quad \sigma = \mathbf{u}^*\mathbf{c},\ \ \alpha = \rho/\sigma \\
\quad \mathbf{r} \leftarrow \mathbf{r} - \alpha\,\mathbf{c} \\
\quad \mathbf{x} \leftarrow \mathbf{x} + \alpha\,\mathbf{u} \\
\texttt{end while}
\end{array}
$$

$\mathbf{A}\mathbf{x} = \mathbf{b}$ $\qquad$ $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$

Subspace methods

Iterative. In each step

- **Expansion.** Expand the search subspace.
- **Extraction.** Extract an appropriate approximate solution from the search subspace.

Example. Krylov subspace methods
as GMRES, CG, Arnoldi, Lanczos: expansion by $\mathbf{t}_k = \mathbf{A}\mathbf{v}_k$

Note. Algorithms may exploit the same basis
for expansion as for extraction (GMRES, CG)
but also different ones (GCR)

Krylov subspace expansion

The columns of $\mathbf{V}_k = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$ form a (orthonormal) Krylov basis: then $\mathbf{t} = \mathbf{A}\mathbf{v}_k$ is an expansion vector.

Examples. For $\mathbf{A}\mathbf{x} = \mathbf{b}$

- GMRES (minimal residual extraction),
- CG for $\mathbf{A}\mathbf{x} = \mathbf{b}$ if $\mathbf{A}^* = \mathbf{A}$ (Galerkin extraction).

For $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$

- **Arnoldi** ((harmonic) Ritz extraction).
- **Shift and Invert Arnoldi** ((harmonic) Ritz extraction): $\mathbf{t} = (\mathbf{A} - \tau\mathbf{I})^{-1}\mathbf{v}_k$ to generate a basis

for the search subspace $\mathcal{K}_k((\mathbf{A} - \tau\mathbf{I})^{-1}, \mathbf{v}_1)$.

Convergence without subspace accelaration

Shift & Invert $\qquad \mathbf{u}_{k+1} = (\mathbf{A} - \tau\mathbf{I})^{-1}\mathbf{u}_k.$ $\qquad$ (s&i)

Then $\qquad \tan\angle(\mathbf{x}, \mathbf{u}_k) \sim \left(\dfrac{\lambda - \tau}{\lambda_j - \tau}\right)^k,$

where $\lambda = \lambda_{j_0}$, $\mathbf{x} = \mathbf{x}_{j_0}$, and $|\lambda_{j_0} - \tau| < |\lambda_j - \tau|$ all $j \neq j_0$.

Rayleigh Quotient Iteration

$$\mathbf{u}_{k+1} = (\mathbf{A} - \rho_k\mathbf{I})^{-1}\mathbf{u}_k, \text{ where } \rho_k \equiv \rho(\mathbf{u}_k).$$

$\mathbf{A}^* = \mathbf{A}$, $\quad \alpha_k \equiv \frac{|\rho_k - \lambda|}{\gamma - |\rho_k - \lambda|}$, $\quad \zeta_k \equiv \tan\angle(\mathbf{u}_k, \mathbf{x})$.

Then $\quad \zeta_{k+1} \le \alpha_k\zeta_k, \quad \alpha_{k+1} \le (\alpha_k\zeta_k)^2$

$\Rightarrow \quad$ asymptotic cubic convergence

(Approximately) solving for $\mathbf{t} \perp \mathbf{u}$

$$(\mathbf{I}-\mathbf{u}\mathbf{u}^*)(\mathbf{A}-\vartheta\mathbf{I})(\mathbf{I}-\mathbf{u}\mathbf{u}^*)\mathbf{t} = -\mathbf{r} \qquad \text{(jd)}$$

If solved exactly $\Rightarrow$ asymptotic quadratic convergence

Approximate solves. $\mathbf{M} \approx \mathbf{A}-\vartheta\mathbf{I}$

$\mathbf{t} \perp \mathbf{u}$ such that $(\mathbf{I}-\mathbf{u}\mathbf{u}^*)\mathbf{M}(\mathbf{I}-\mathbf{u}\mathbf{u}^*)\mathbf{t} = -\mathbf{r}$

$\Leftrightarrow$ $\mathbf{t} = -(\mathbf{I}-\frac{\mathbf{w}\,\mathbf{u}^*}{\mathbf{u}^*\mathbf{w}})\mathbf{M}^{-1}\mathbf{r}$, where $\mathbf{w} \equiv \mathbf{M}^{-1}\mathbf{u}$

Expansion by $\mathbf{t} = -\mathbf{M}^{-1}\mathbf{r}$ (d) **Davidson** '75

Expansion by $\mathbf{t} = -(\mathbf{I}-\frac{\mathbf{w}\,\mathbf{u}^*}{\mathbf{u}^*\mathbf{w}})\mathbf{M}^{-1}\mathbf{r}$ (o) **Olsen** '93

(Approximately) solving for $\mathbf{t} \perp \mathbf{u}$

$$(\mathbf{I}-\mathbf{u}\mathbf{u}^*)(\mathbf{A}-\vartheta\mathbf{I})(\mathbf{I}-\mathbf{u}\mathbf{u}^*)\mathbf{t} = -\mathbf{r} \qquad \text{(jd)}$$

If solved exactly $\Rightarrow$ asymptotic quadratic convergence

Approximate solves. $\mathbf{M} \approx \mathbf{A}-\vartheta\mathbf{I}$

$\mathbf{t} \perp \mathbf{u}$ such that $(\mathbf{I}-\mathbf{u}\mathbf{u}^*)\mathbf{M}(\mathbf{I}-\mathbf{u}\mathbf{u}^*)\mathbf{t} = -\mathbf{r}$

$\Leftrightarrow$ $\mathbf{t} = -(\mathbf{I}-\frac{\mathbf{w}\,\mathbf{u}^*}{\mathbf{u}^*\mathbf{w}})\mathbf{M}^{-1}\mathbf{r}$, where $\mathbf{w} \equiv \mathbf{M}^{-1}\mathbf{u}$

Precondition (jd) by $(\mathbf{I}-\frac{\mathbf{w}\,\mathbf{u}^*}{\mathbf{u}^*\mathbf{w}})\mathbf{M}^{-1}$, where $\mathbf{w} \equiv \mathbf{M}^{-1}\mathbf{u}$

With $\mathbf{x}_0 = \mathbf{0}$, **preconditoned Krylov** requires mult. by

$$(\mathbf{I}-\frac{\mathbf{w}\,\mathbf{u}^*}{\mathbf{u}^*\mathbf{w}})\mathbf{M}^{-1}(\mathbf{A}-\vartheta\mathbf{I})$$

with righthand side vector $(\mathbf{I}-\frac{\mathbf{w}\,\mathbf{u}^*}{\mathbf{u}^*\mathbf{w}})\mathbf{M}^{-1}\mathbf{r}$.

Additional costs. (additional to RQI)
per step mult. by $\mathbf{I}-\frac{\mathbf{w}\,\mathbf{u}^*}{\mathbf{u}^*\mathbf{w}}$: 1 AXPY, 1 DOT per step
per Krylov run: 1 solve of $\mathbf{M}\mathbf{w} = \mathbf{u}$.

Jacobi–Davidson

- Subspace method
  - \+ Accelerated convergence
  - \+ Steering possibilities
  - \+ variety of selection methods
  - − More costly steps
- Expansion vectors from JD equation
  - \+ Locally optimal expansion (with exact solves)
  - \+ Asymptotic quadratic convergence possible (with exact solves)
  - \+ Well-conditioned (when λ is simple)
  - \+ Fast convergence with moderate accurate solves
  - \+ Preconditioners can be exploited
  - − Additional costs per step

Restart

If $\mathbf{V}$ is $n \times k$ and $k > k_{\max}$

With $k = k_{\min}$,

- ∘ select k promising approximate eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_{\min}$,
- ∘ continu with $\mathbf{V} = [\mathbf{u}_1, \ldots, \mathbf{u}_k]$.

Example. Select $y_1, \ldots, y_k$ pre (harmonic) Ritz vectors with (harmonic) Ritz value closest to the target τ:

$$\mathbf{V} \leftarrow \mathbf{V}Y \text{ with } Y = [y_1, \ldots, y_k].$$

Example. Select $y_1, \ldots, y_{k-1}$ pre (harmonic) Ritz vectors with (harmonic) Ritz value closest to the target τ, select best pre (harmonic) Ritz vector y_1' wrt $[\mathbf{v}_1, \ldots, \mathbf{v}_{k-1}]$,
$\mathbf{V} \leftarrow \mathbf{V}Y$ with Y orthogonal version $[({y_1'}^{\mathsf{T}}, 0)^{\mathsf{T}}, y_1, \ldots, y_k]$.

Utrecht, 7 december 2011

Implicitly Restarted Arnoldi

Gerard Sleijpen

Universiteit Utrecht
Department of Mathematics

http://www.staff.science.uu.nl/~sleij101/

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

with $\mathbf{A}$ $n \times n$.

Today's topic.

- Arnoldi's method and extensions.

Program

- Arnoldi and polynomial filtering
- Polynomials and Ritz pairs
- Implicitly Restarted Arnoldi
- Upon convergence
- Rational Krylov sequence

Arnoldi and polynomial filtering

Suppose we have the Arnoldi relation $\mathbf{A}\mathbf{V}_m = \mathbf{V}_{m+1}\,\underline{H}_m$.

$\mathcal{K}_{m+1}(\mathbf{A}, \mathbf{v}_1)$ contains all vectors $p(\mathbf{A})\mathbf{v}_1$ where $p \in \mathcal{P}_m$.

$\mathcal{P}_m$ is the space of all polynomials of degree at most m.

How to compute $\mathbf{w} = p(\mathbf{A})\mathbf{v}_1$ for specific polynomials? (stability, efficiency)

How to select appropriate polynomials?

Application. If m is large (larger than the number of wanted eigenpairs), restart with a $\mathbf{w}$ that is 'rich' in the wanted eigenvector components.