# AMS526: Numerical Analysis I (Numerical Linear Algebra)

## Lecture 23: Arnoldi/Lanczos Iterations Conjugate Gradient Method

Xiangmin Jiao

SUNY Stony Brook

December 4, 2008

# Outline

# Krylov Subspace Methods

- Given $\boldsymbol{A}$ and $\boldsymbol{b}$, Krylov subspace

$$\mathcal{K}_n = \{\boldsymbol{b}, \boldsymbol{Ab}, \boldsymbol{A}^2\boldsymbol{b}, \ldots, \boldsymbol{A}^{n-1}\boldsymbol{b}\}$$

| | linear systems | eigenvalue problems |
|---|---|---|
| Hermitian | CG | Lanczos |
| Nonhermitian | GMRES, BiCG, etc. | Arnoldi |

# Arnoldi Algorithm

- Let $\boldsymbol{Q}_n = [\boldsymbol{q}_1 \mid \boldsymbol{q}_2 \mid \cdot \mid \boldsymbol{q}_n]$ be $m \times n$ matrix with first $n$ columns of $\boldsymbol{Q}$ and $\tilde{\boldsymbol{H}}_n$ be $(n+1) \times n$ upper-left section of $\boldsymbol{H}$
- Start by picking a random $\boldsymbol{q}_1$ and then determine $\boldsymbol{q}_2$ and $\tilde{\boldsymbol{H}}_1$
- The $n$th columns of $\boldsymbol{A}\boldsymbol{Q}_n = \boldsymbol{Q}_{n+1}\tilde{\boldsymbol{H}}_n$ can be written as

$$\boldsymbol{A}\boldsymbol{q}_n = h_{1n}\boldsymbol{q}_1 + \cdots + h_{nn}\boldsymbol{q}_n + h_{n+1,n}\boldsymbol{q}_{n+1}$$

Algorithm: Arnoldi Iteration

given random nonzero $\boldsymbol{b}$, let $\boldsymbol{q}_1 = \boldsymbol{b}/\|\boldsymbol{b}\|$
**for** $n = 1$ **to** $1, 2, 3, \dots$
  $\boldsymbol{v} = \boldsymbol{A}\boldsymbol{q}_n$
  **for** $j = 1$ **to** $n$
    $h_{jn} = \boldsymbol{q}_j^* \boldsymbol{v}$
    $\boldsymbol{v} = \boldsymbol{v} - h_{jn}\boldsymbol{q}_j$
  $h_{n+1,n} = \|\boldsymbol{v}\|$
  $\boldsymbol{q}_{n+1} = \boldsymbol{v}/h_{n+1,n}$

# Lanczos Iteration for Symmetric Matrices

- For symmetric $\boldsymbol{A}$, $\tilde{\boldsymbol{H}}_n$ and $\boldsymbol{H}_n$ are tridiagonal, denoted by $\tilde{\boldsymbol{T}}_n$ and $\boldsymbol{T}_n$, respectively. $\boldsymbol{A}\boldsymbol{Q}_n = \boldsymbol{Q}_{n+1}\tilde{\boldsymbol{H}}_n$ can be written as three-term recurrence

$$\boldsymbol{A}\boldsymbol{q}_n = \beta_{n-1}\boldsymbol{q}_{n-1} + \alpha_n\boldsymbol{q}_n + \beta_n\boldsymbol{q}_{n+1}$$

  where $\alpha_i$ are diagonal entries and $\beta_i$ are sub-diagonal entries of $\tilde{\boldsymbol{T}}_n$

```
Algorithm: Lanczos Iteration
  β_0 = 0, q_0 = 0
  given random b, let q_1 = b/||b||
  for n = 1 to 1, 2, 3, ...
        v = A q_n
        α_n = q_n v
        v = v − β_{n−1} q_{n−1} − α_n q_n
        β_n = ||v||
        q_{n+1} = v/β_n
```

Question: What are the meanings of $\alpha_n$ and $\beta_n$ in $\tilde{\boldsymbol{T}}_n$?

# Properties of Arnoldi and Lanczos Iterations

- Eigenvalues of $\boldsymbol{H}_n$ (or $\boldsymbol{T}_n$ in Lanczos iterations) are called *Ritz values.*
- When $m = n$, Ritz values are eigenvalues.
- Even for $n \ll m$, Ritz values are often accurate approximations to eigenvalues of $\boldsymbol{A}$!
- For symmetric matrices with evenly spaced eigenvalues, Ritz values tend to first convert to extreme eigenvalue.
- With rounding errors, Lanczos iteration can suffer from loss of orthogonality and can in turn lead to spurious "ghost" eigenvalues.

# Arnoldi and Polynomial Approximation

- For any $\boldsymbol{x} \in \mathcal{K}_{n+1} = \{\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \boldsymbol{A}^2\boldsymbol{b}, \ldots, \boldsymbol{A}^n\boldsymbol{b}\}$,

$$\boldsymbol{x} = c_0\boldsymbol{b} + c_1\boldsymbol{A}\boldsymbol{b} + c_2\boldsymbol{A}^2\boldsymbol{b} + \cdots + c_n\boldsymbol{A}^n\boldsymbol{b}.$$

- It can be interpreted as a polynomial in $\boldsymbol{A}$ times $\boldsymbol{b}$, $\boldsymbol{x} = p(\boldsymbol{A})\boldsymbol{b}$, where

$$p(z) = c_0 + c_1 z + c_2 z^2 + \cdots + c_n z^n.$$

- Krylov subspace iterations are often analyzed in terms of matrix polynomials.
- Let $P^n$ be the set of polynomials of degree $n$ with $c_n = 1$.
- Optimality of Arnoldi/Lanczos approximation: It finds $p^* \in P^n$ s.t.
  - $p^*$ is optimal polynomial among $p \in P^n$ that minimizes $\|p(\boldsymbol{A})\boldsymbol{b}\|$, which is equivalent to minimizing the distance betwen $\boldsymbol{A}^n\boldsymbol{b}$ and its projection in $\mathcal{K}_n$
  - $p^*$ is the characteristic polynomial of $\boldsymbol{H}_n$
- Ritz values are the roots of an optimal polynomial

# Outline

# Krylov Subspace Algorithms

- Create a sequence of Krylov subspaces for $\boldsymbol{Ax} = \boldsymbol{b}$

$$\mathcal{K}_n = \{\boldsymbol{b}, \boldsymbol{Ab}, \dots, \boldsymbol{A}^{n-1}\boldsymbol{b}\}$$

  and find an approximate (hopefully optimal) solutions $\boldsymbol{x}_n$ in $\mathcal{K}_n$
- Only matrix-vector products involved
- For SPD matrices, most famous algorithm is *Conjugate Gradient* (*CG*) method discovered by Hestenes/Stiefel in 1952
  - Finds best solution $\boldsymbol{x}_n \in \mathcal{K}_n$ in norm $\|\boldsymbol{x}\|_A = \sqrt{\boldsymbol{x}^T\boldsymbol{Ax}}$
  - Only requires storing 4 vectors (instead of $n$ vectors) due to three-term recurrence

# Motivation of Conjugate Gradients

- If $\boldsymbol{A}$ is $m \times m$ SPD, then quadratic function
$$\varphi(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{b}$$
has unique minimum
- Negative gradient of this function is residual vector
$$-\nabla\varphi(\boldsymbol{x}) = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x} = \boldsymbol{r}$$
so minimum is obtained precisely when $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$
- Optimization methods have form
$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \alpha\boldsymbol{p}_n$$
where $\boldsymbol{p}_n$ is *search direction* and $\alpha$ is *step length* chosen to minimize $\varphi(\boldsymbol{x}_n + \alpha\boldsymbol{p}_n)$
- Line search parameter can be determined analytically as $\alpha = \boldsymbol{r}_n^T\boldsymbol{p}_n / \boldsymbol{p}_n^T\boldsymbol{A}\boldsymbol{p}_n$
- In CG, $\boldsymbol{p}_n$ is chosen to be A-conjugate (or A-orthogonal) to previous search directions, i.e., $\boldsymbol{p}_n^T\boldsymbol{A}\boldsymbol{p}_j = 0$ for $j < n$

# Conjugate Gradient Method

Algorithm: Conjugate Gradient Method

$\boldsymbol{x}_0 = 0$, $\boldsymbol{r}_0 = \boldsymbol{b}$, $\boldsymbol{p}_0 = \boldsymbol{r}_0$

**for** $n = 1$ **to** $1, 2, 3, \dots$

| | |
|---|---|
| $\alpha_n = (\boldsymbol{r}_{n-1}^T \boldsymbol{r}_{n-1})/(\boldsymbol{p}_{n-1}^T \boldsymbol{A}\boldsymbol{p}_{n-1})$ | step length |
| $\boldsymbol{x}_n = \boldsymbol{x}_{n-1} + \alpha_n \boldsymbol{p}_{n-1}$ | approximate solution |
| $\boldsymbol{r}_n = \boldsymbol{r}_{n-1} - \alpha_n \boldsymbol{A}\boldsymbol{p}_{n-1}$ | residual |
| $\beta_n = (\boldsymbol{r}_n^T \boldsymbol{r}_n)/(\boldsymbol{r}_{n-1}^T \boldsymbol{r}_{n-1})$ | improvement this step |
| $\boldsymbol{p}_n = \boldsymbol{r}_n + \beta_n \boldsymbol{p}_{n-1}$ | search direction |

- Only one matrix-vector product $\boldsymbol{A}\boldsymbol{p}_{n-1}$ per iteration
- Apart from matrix-vector product, operation count per iteration is $O(m)$
- If $\boldsymbol{A}$ is sparse with constant number of nonzeros per row, $O(m)$ operations per iteration.