AMS526: Numerical Analysis I
(Numerical Linear Algebra)
Lecture 22: More on Conjugate Gradient Method; Preconditioned Conjugate Gradient Method

Xiangmin Jiao

Stony Brook University
An Alternative Interpretation of CG

<table>
<thead>
<tr>
<th>Algorithm: CG</th>
<th>Algorithm: A non-standard CG</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_0 = 0$, $r_0 = b$, $p_0 = r_0$</td>
<td>$x_0 = 0$, $r_0 = b$, $p_0 = r_0$</td>
</tr>
<tr>
<td><strong>for</strong> $n = 1$ <strong>to</strong> $1, 2, 3, \ldots$</td>
<td><strong>for</strong> $n = 1$ <strong>to</strong> $1, 2, 3, \ldots$</td>
</tr>
<tr>
<td>$\alpha_n = r_{n-1}^T r_{n-1} / (p_{n-1}^T A p_{n-1})$</td>
<td>$\alpha_n = r_{n-1}^T p_{n-1} / (p_{n-1}^T A p_{n-1})$</td>
</tr>
<tr>
<td>$x_n = x_{n-1} + \alpha_n p_{n-1}$</td>
<td>$x_n = x_{n-1} + \alpha_n p_{n-1}$</td>
</tr>
<tr>
<td>$r_n = r_{n-1} - \alpha_n A p_{n-1}$</td>
<td>$r_n = b - A x_n$</td>
</tr>
<tr>
<td>$\beta_n = r_n^T r_n / (r_{n-1}^T r_{n-1})$</td>
<td>$\beta_n = -r_{n-1}^T A p_{n-1} / (p_{n-1}^T A p_{n-1})$</td>
</tr>
<tr>
<td>$p_n = r_n + \beta_n p_{n-1}$</td>
<td>$p_n = r_n + \beta_n p_{n-1}$</td>
</tr>
</tbody>
</table>

- The non-standard one is less efficient but easier to understand
- It is easy to see $r_n = r_{n-1} - \alpha_n A p_{n-1} = b - A x_n$
- We need to show:
  - $\alpha_n$ minimizes $\varphi$ along search direction $p_n$
  - $\alpha_n$ and $\beta_n$ are equivalent to those in standard CG
  - Minimizing $\varphi$ along $p_n$ also minimizes $\varphi$ within Krylov subspace
Optimality of Step Length

- Select step length $\alpha_n$ over vector $p_{n-1}$ to minimize
  \[ \varphi(x) = \frac{1}{2} x^T A x - x^T b \]
- Let $x_n = x_{n-1} + \alpha_n p_{n-1}$,
  \[
  \varphi(x_n) = \frac{1}{2} (x_{n-1} + \alpha_n p_{n-1})^T A (x_{n-1} + \alpha_n p_{n-1}) - (x_{n-1} + \alpha_n p_{n-1})^T b
  \]
  \[
  = \frac{1}{2} \alpha_n^2 p_{n-1}^T A p_{n-1} + \alpha_n p_{n-1}^T A x_{n-1} - \alpha_n p_{n-1}^T b + \text{constant}
  \]
  \[
  = \frac{1}{2} \alpha_n^2 p_{n-1}^T A p_{n-1} - \alpha_n p_{n-1}^T r_{n-1} + \text{constant}
  \]

- Therefore,
  \[
  \frac{d\varphi}{d\alpha_n} = 0 \Rightarrow \alpha_n p_{n-1}^T A p_{n-1} - p_{n-1}^T r_{n-1} = 0 \Rightarrow \alpha_n = \frac{p_{n-1}^T r_{n-1}}{p_{n-1}^T A p_{n-1}}.
  \]

- In addition, $p_{n-1}^T r_{n-1} = r_{n-1}^T r_{n-1}$ because $p_{n-1} = r_{n-1} + \beta_n p_{n-2}$ and $r_{n-1}^T p_{n-2} = 0$ due to the following theorem.
Outline

1. Properties of Conjugate Gradient

2. Preconditioned Conjugate Gradient
Properties of Conjugate Gradients

Theorem

If \( r_{n-1} \neq 0 \), spaces spanned by approximate solutions \( x_n \), search directions \( p_n \), and residuals \( r_n \) are all equal to Krylov subspaces

\[
K_n = \langle x_1, x_2, \ldots, x_n \rangle = \langle p_0, p_1, \ldots, p_{n-1} \rangle = \langle r_0, r_1, \ldots, r_{n-1} \rangle = \langle b, Ab, \ldots, A^{n-1}b \rangle
\]

The residuals are orthogonal (i.e., \( r_n^T r_j = 0 \) for \( j < n \)) and search directions are A-conjugate (i.e., \( p_n^T A p_j = 0 \) for \( j < n \)).

This theorem implies that

\[
\alpha_n = \frac{(r_{n-1}^T r_{n-1})}{(p_{n-1}^T A p_{n-1})} = r_{n-1}^T p_{n-1}/(p_{n-1}^T A p_{n-1})
\]

and

\[
\beta_n = \frac{r_n^T r_n}{r_{n-1}^T r_{n-1}} = \frac{r_n^T (r_{n-1} - \alpha_n A p_{n-1})}{r_{n-1}^T r_{n-1}} = -\frac{r_n^T A p_{n-1}}{p_{n-1}^T A p_{n-1}}.
\]
Proof of Properties of CG

Prove based on notation of standard CG.

- Proof of equality of subspaces by simple induction.
- To prove $r_n^T r_j = 0$, note that $r_n = r_{n-1} - \alpha_n A p_{n-1}$ and $(A p_{n-1})^T = p_{n-1}^T A$, so

  $$r_n^T r_j = (r_{n-1} - \alpha_n A p_{n-1})^T r_j = r_{n-1}^T r_j - \alpha_n p_{n-1}^T A r_j.$$

  - If $j < n - 1$, then both terms on right are zero by induction.
  - If $j = n - 1$, plug in $\alpha_n = (r_{n-1}^T r_{n-1})/(p_{n-1}^T A p_{n-1})$

    $$r_{n-1}^T r_j - \alpha_n p_{n-1}^T A r_j = r_{n-1}^T r_{n-1} - r_{n-1}^T r_{n-1} \frac{p_{n-1}^T A r_{n-1}}{p_{n-1}^T A p_{n-1}},$$

    which is zero because

    $$p_{n-1}^T A p_{n-1} = p_{n-1}^T A (r_{n-1} + \beta_n p_{n-2}) = p_{n-1}^T A r_{n-1}$$

    by induction hypothesis.
To prove $p_n^T A p_j = 0$, note that $p_n = r_n + \beta_n p_{n-1}$, so

$$p_n^T A p_j = r_n^T A p_j + \beta_n p_{n-1}^T A p_j.$$

- If $j < n - 1$, then both terms on right are zero by induction.
- If $j = n - 1$, plug in $\beta_n = (r_n^T r_n)/(r_{n-1}^T r_{n-1})$,

$$r_n^T A p_j + \beta_n p_{n-1}^T A p_j = r_n^T A p_{n-1} + \frac{1}{\alpha_n} r_n^T r_n$$

$$= \frac{1}{\alpha_n} r_n^T (r_n + \alpha_n A p_{n-1})$$

$$= \frac{1}{\alpha_n} r_n^T r_{n-1}$$

$$= 0.$$
Optimality of Conjugate Gradients

**Theorem**

If \( r_{n-1} \neq 0 \), then error \( e_n = x_\ast - x_n \) are minimized in A-norm in \( \mathcal{K}_n \).

**Proof.**

Consider arbitrary point \( x = x_n - \Delta x \in \mathcal{K}_n \) with error \( e = x_\ast - x = e_n + \Delta x \). So

\[
\|e\|^2_A = (e_n + \Delta x)^T A (e_n + \Delta x) = e_n^T A e_n + \Delta x^T A \Delta x + 2 e_n^T A \Delta x,
\]

where \( e_n^T A \Delta x = r_n^T \Delta x = 0 \) because \( r_n \perp \mathcal{K}_n \). Since \( A \) is SPD, \( \|e\|^2_A \geq \|e_n\|^2_A \) and equality holds iff \( \Delta x = 0 \).

Because \( \mathcal{K}_n \) grows monotonically, error decreases monotonically.
Relationship with Lanczos Iteration

- In conjugate gradient,

\[ \mathcal{K}_n = \langle x_1, x_2, \ldots, x_n \rangle = \langle p_0, p_1, \ldots, p_{n-1} \rangle = \langle r_0, r_1, \ldots, r_{n-1} \rangle = \langle b, Ab, \ldots, A^{n-1}b \rangle, \]

where \( b \) is right-hand side of \( Ax = b \)

- In Lanczos Iteration (Arnoldi algorithm for symmetric matrices),

\[ \mathcal{K}_n = \langle b, Ab, \ldots, A^{n-1}b \rangle = \langle q_1, q_2, \ldots, q_n \rangle \]

where \( b \) is chosen arbitrarily

- If \( q_1 \) is a multiple of \( r_0 \) then \( q_i \) will be proportional to \( r_{i-1} \)

Therefore, conjugate gradient and Lanczos iteration are essentially the same process, so it is possible to obtain information about spectrum of \( A \) within CG algorithm
Rate of Convergence

- In addition, CG can be studied in terms of polynomial approximation
  - It finds optimal polynomial $p_n \in P_n$ of degree $n$ with $p(0) = 1$, minimizing $\|p_n(A)e_0\|_A$ with initial error $e_0 = x^*$
  - Convergence results can be obtained from this polynomial approximation

- Some important convergence results
  - If $A$ has $n$ distinct eigenvalues, CG converges in at most $n$ steps
  - If $A$ has 2-norm condition number $\kappa$, the errors are
    \[
    \frac{\|e_n\|_A}{\|e_0\|_A} \leq 2 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^n
    \]
    which is $\approx 2 \left( 1 - \frac{2}{\sqrt{\kappa}} \right)^n$ as $\kappa \to \infty$. So convergence is expected in $O(\sqrt{\kappa})$ iterations.

- In general, CG performs well with clustered eigenvalues
Outline

1. Properties of Conjugate Gradient

2. Preconditioned Conjugate Gradient
Preconditioning

- Motivation: Convergence of iterative methods heavily depends on eigenvalues or singular values of equation
- Main idea of preconditioning is to introduce a nonsingular matrix $M$ such that $M^{-1}A$ has better properties than $A$. Thereafter, solve

$$M^{-1}Ax = M^{-1}b,$$

which has the same solution as $Ax = b$

- Criteria of $M$
  - “Good” approximation of $A$, depending on iterative solvers
  - Ease of inversion

- Typically, a precondition $M$ is good if $M^{-1}A$ is not too far from normal and its eigenvalues are clustered
Left, Right, and Hermitian Preconditioners

- Left preconditioner: Left multiply $M^{-1}$ and solve $M^{-1}Ax = M^{-1}b$
- Right preconditioner: Right multiply $M^{-1}$ and solve $AM^{-1}y = b$ with $x = M^{-1}y$
- However, if $A$ is Hermitian, $M^{-1}A$ or $AM^{-1}$ breaks symmetry
- How to resolve this problem?
Left, Right, and Hermitian Preconditioners

- Left preconditioner: Left multiply $M^{-1}$ and solve $M^{-1}Ax = M^{-1}b$
- Right preconditioner: Right multiply $M^{-1}$ and solve $AM^{-1}y = b$ with $x = M^{-1}y$
- However, if $A$ is Hermitian, $M^{-1}A$ or $AM^{-1}$ breaks symmetry
- How to resolve this problem?

- Suppose $M$ is Hermitian positive definite, with $M = CC^*$ for some $C$, then $Ax = b$ is equivalent to

$$\begin{bmatrix} C^{-1}AC^{-*} \end{bmatrix} (C^*x) = C^{-1}b,$$

where $C^{-1}AC^{-*}$ is Hermitian positive definite, and it is similar to $C^{-*}C^{-1}A = M^{-1}A$ and has same eigenvalues as $M^{-1}A$
- Example of $M = CC^*$ is Cholesky factorization $M = RR^*$, where $R$ is upper triangular
Preconditioned Conjugate Gradient

- When preconditioning a symmetric matrix, use SPD matrix $M$, and $M = RR^T$
- In practice, algorithm can be organized so that only $M^{-1}$ (instead of $R^{-1}$) appears

Algorithm: Preconditioned Conjugate Gradient Method

\[
x_0 = 0, \quad r_0 = b, \quad p_0 = M^{-1}r_0, \quad z_0 = p_0
\]

for $n = 1, 2, 3, \ldots$

$\alpha_n = (r_{n-1}^Tz_{n-1})/(p_{n-1}^TAp_{n-1})$  
step length

$x_n = x_{n-1} + \alpha_n p_{n-1}$  
approximate solution

$r_n = r_{n-1} - \alpha_n Ap_{n-1}$  
residual

$z_n = M^{-1}r_n$  
preconditioning

$\beta_n = (r_n^Tz_n)/(r_{n-1}^Tz_{n-1})$  
improvement this step

$p_n = z_n + \beta_n p_{n-1}$  
search direction
Effective Preconditioners for CG

- **SSOR Preconditioner**
  - Simpler form: use matrix splitting of form $A = L + D + L^T$ and take
    
    \[ M = (D + L)D^{-1}(D + L)^T \]
  - More generally, introduce SSOR relaxation parameter $\omega$, and take
    
    \[ M = \frac{1}{2 - \omega} \left( \frac{1}{\omega} D + L \right) \left( \frac{1}{\omega} D \right)^{-1} \left( \frac{1}{\omega} D + L \right)^T. \]

    With optimal $\omega$, $\text{cond}(M^{-1}A) = O(\sqrt{\text{cond}(A)})$, but determining optimal $\omega$ is impractical

- **Incomplete factorization**
  - If $A = LL^T$ were used as preconditioner, then $\text{cond}(M^{-1}A) = 1$, but impractical
  - Instead, compute approximate factorization $A \approx \tilde{L}\tilde{L}^T$, which omit all fills or omit small fills and use $M = \tilde{L}\tilde{L}^T$ as preconditioner
Other Commonly Used Preconditioners

- **Jacobi preconditioning**: $M = \text{diag}(A)$. Very simple and cheap, might improve certain problems but usually insufficient.

- **Block-Jacobi preconditioning**: Let $M$ be composed of block-diagonal instead of diagonal.

- **Multigrid (coarse-grid approximations)**: For a PDE discretized on a grid, a preconditioner can be formed by transferring the solution to a coarser grid, solving a smaller problem, then transferring back. This is sometimes the most efficient approach if applicable.