AMS526: Numerical Analysis I (Numerical Linear Algebra for Computational and Data Sciences)

Lecture 22: Conjugate Gradient Method

Xiangmin Jiao

SUNY Stony Brook

Outline

1. CG as Optimization Method
2. CG and Krylov Subspace
3. Convergence Properties of CG

Krylov Subspace Algorithms

- Create a sequence of Krylov subspaces for $Ax = b$

  $$\mathcal{K}_k = \{b, Ab, \dots, A^{k-1}b\}$$

  and find an "optimal" solutions x_k in $\mathcal{K}_k$ at kth step
- Only matrix-vector products involved
- For SPD matrices, the most famous method is *Conjugate Gradient* (*CG*) method discovered by Hestenes/Stiefel in 1952
  - Finds best solution $x_k \in \mathcal{K}_k$ in norm $\|x\|_A \equiv \sqrt{x^T Ax}$
  - Only requires storing 4 vectors (instead of k vectors) due to three-term recurrence

Motivation of Conjugate Gradients

- If $A \in \mathbb{R}^{n \times n}$ is SPD, then quadratic function

$$\varphi(x) = \frac{1}{2} x^T A x - x^T b$$

  has unique minimum

- Negative gradient of this function is residual vector

$$-\nabla \varphi(x) = b - Ax = r$$

  so minimum is obtained precisely when $Ax = b$

- Optimization methods have form

$$x_{k+1} = x_k + \alpha_k p_k$$

  where p_k is *search direction* and α is *step length* chosen to minimize $\varphi(x_k + \alpha_k p_k)$

- Line search parameter is $\alpha_k = r_k^T p_k / p_k^T A p_k$
- In CG, p_k is chosen to be A-conjugate (or A-orthogonal) to previous search directions, i.e., $p_k^T A p_j = 0$ for $j < k$

Conjugate Gradient Method

```
Algorithm: Conjugate Gradient Method
  $x_0 = 0,\ r_0 = b,\ p_0 = r_0$
  for $k = 1, 2, 3, \ldots$
      $\alpha_k = (r_{k-1}^T r_{k-1})/(p_{k-1}^T A p_{k-1})$        step length
      $x_k = x_{k-1} + \alpha_k p_{k-1}$                              approximate solution
      $r_k = r_{k-1} - \alpha_k A p_{k-1}$                            residual
      $\beta_k = (r_k^T r_k)/(r_{k-1}^T r_{k-1})$                     improvement this step
      $p_k = r_k + \beta_k p_{k-1}$                                   search direction
```

- Only one matrix-vector product Ap_{k-1} per iteration
- Apart from matrix-vector product, #flops per iteration is $O(n)$
- If A is sparse with constant number of nonzeros per row, $O(n)$ operations per iteration
- CG can be viewed as minimization of quadratic function $\varphi(x) = \frac{1}{2}x^T Ax - x^T b$ by modifying steepest descent

An Alternative Interpretation of CG

```
Algorithm: CG
    $x_0 = 0,\ r_0 = b,\ p_0 = r_0$
    for $k = 1, 2, 3, \dots$
        $\alpha_k = (r_{k-1}^T r_{k-1})/(p_{k-1}^T A p_{k-1})$
        $x_k = x_{k-1} + \alpha_k p_{k-1}$
        $r_k = r_{k-1} - \alpha_k A p_{k-1}$
        $\beta_k = (r_k^T r_k)/(r_{k-1}^T r_{k-1})$
        $p_k = r_k + \beta_k p_{k-1}$
```

```
Algorithm: A non-standard CG
    $x_0 = 0,\ r_0 = b,\ p_0 = r_0$
    for $k = 1, 2, 3, \dots$
        $\alpha_k = r_{k-1}^T p_{k-1}/(p_{k-1}^T A p_{k-1})$
        $x_k = x_{k-1} + \alpha_k p_{k-1}$
        $r_k = b - A x_k$
        $\beta_k = -r_k^T A p_{k-1}/(p_{k-1}^T A p_{k-1})$
        $p_k = r_k + \beta_k p_{k-1}$
```

- The non-standard one is less efficient but easier to understand
- It is easy to see $r_k = r_{k-1} - \alpha_k A p_{k-1} = b - A x_k$
- We need to show:
    - α_k minimizes φ along search direction p_k
    - α_k and β_k are equivalent to those in standard CG
    - Minimizing φ along p_k also minimizes φ within Krylov subspace

Optimality of Step Length

- Select step length α_k over vector p_{k-1} to minimize $\varphi(x) = \frac{1}{2}x^T Ax - x^T b$
- Let $x_k = x_{k-1} + \alpha_k p_{k-1}$,

$$\begin{aligned}\varphi(\alpha_k) &= \frac{1}{2}(x_{k-1} + \alpha_k p_{k-1})^T A(x_{k-1} + \alpha_k p_{k-1}) - (x_{k-1} + \alpha_k p_{k-1})^T b \\ &= \frac{1}{2}\alpha_k^2 p_{k-1}^T A p_{k-1} + \alpha_k p_{k-1}^T A x_{k-1} - \alpha_k p_{k-1}^T b + \text{constant} \\ &= \frac{1}{2}\alpha_k^2 p_{k-1}^T A p_{k-1} - \alpha_k p_{k-1}^T r_{k-1} + \text{constant}\end{aligned}$$

- Therefore,

$$\frac{d\varphi}{d\alpha_k} = 0 \Rightarrow \alpha_k p_{k-1}^T A p_{k-1} - p_{k-1}^T r_{k-1} = 0 \Rightarrow \alpha_k = \frac{p_{k-1}^T r_{k-1}}{p_{k-1}^T A p_{k-1}}.$$

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

Outline

1. CG as Optimization Method

2. CG and Krylov Subspace

3. Convergence Properties of CG

Krylov Subspace in Conjugate Gradient

Theorem (Theorem 38.1 in NLA p. 295)

If $r_{k-1} \neq 0$, spaces spanned by approximate solutions x_k, search directions p_k, and residuals r_k are all equal to Krylov subspaces

$$\begin{aligned}\mathcal{K}_k &= \langle x_1, x_2, \ldots, x_k \rangle = \langle p_0, p_1, \ldots, p_{k-1} \rangle \\ &= \langle r_0, r_1, \ldots, r_{k-1} \rangle = \langle b, Ab, \ldots, A^{k-1} b \rangle\end{aligned}$$

The residuals are orthogonal (i.e., $r_k^T r_j = 0$ for $j < k$) and search directions are A-conjugate (i.e, $p_k^T A p_j = 0$ for $j < k$).

This theorem implies that

$$\alpha_k = (r_{k-1}^T r_{k-1})/(p_{k-1}^T A p_{k-1}) = r_{k-1}^T p_{k-1}/(p_{k-1}^T A p_{k-1})$$

and

$$\beta_k = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}} = \frac{r_k^T (r_{k-1} - \alpha_k A p_{k-1})}{r_{k-1}^T r_{k-1}} = -\frac{r_k^T A p_{k-1}}{p_{k-1}^T A p_{k-1}}.$$

Proof of Properties of CG

Prove based on notation of standard CG.

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

$$r_k^T r_j = (r_{k-1} - \alpha_k A p_{k-1})^T r_j = r_{k-1}^T r_j - \alpha_k p_{k-1}^T A r_j.$$

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

$$r_{k-1}^T r_j - \alpha_k p_{k-1}^T A r_j = r_{k-1}^T r_{k-1} - r_{k-1}^T r_{k-1} \frac{p_{k-1}^T A r_{k-1}}{p_{k-1}^T A p_{k-1}},$$

which is zero because

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

by induction hypothesis.

Proof Cont'd

- To prove $p_k^T A p_j = 0$, note that $p_k = r_k + \beta_k p_{k-1}$, so

$$p_k^T A p_j = r_k^T A p_j + \beta_k p_{k-1}^T A p_j.$$

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

$$\begin{aligned} r_k^T A p_j + \beta_k p_{k-1}^T A p_j &= r_k^T A p_{k-1} + \frac{1}{\alpha_k} r_k^T r_k \\ &= \frac{1}{\alpha_k} r_k^T (r_k + \alpha_k A p_{k-1}) \\ &= \frac{1}{\alpha_k} r_k^T r_{k-1} \\ &= 0. \end{aligned}$$

Relationship with Lanczos Iteration

CG and Lanczos iteration are essentially the same process

- In CG, let b be right-hand side of $Ax = b$

$$\begin{aligned}\mathcal{K}_k &= \langle x_1, x_2, \ldots, x_k \rangle = \langle p_0, p_1, \ldots, p_{k-1} \rangle \\ &= \langle r_0, r_1, \ldots, r_{k-1} \rangle = \langle b, Ab, \ldots, A^{k-1} b \rangle\end{aligned}$$

- In Lanczos iteration for $A \in \mathbb{R}^{n \times n}$, starting from $q_1 = b/\|b\|$

$$AQ_k = Q_{k+1} \tilde{T}_k, \tag{1}$$

  where $\tilde{T}_k$ is $(k+1) \times k$; Q_k is composed of orthonormal basis of $\mathcal{K}_k$
- If q_1 is a multiple of $r_0 = b$, then q_i will be proportional to r_{i-1}
- In (1), $\tilde{T}_k = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{k-1} \\ & & \beta_{k-1} & \alpha_k \\ & & & \beta_k \end{bmatrix}$

Alternative Derivation Based on Lanczos Iteration

- Let $x_k = Q_k y_k$. Then,

$$r_k = b - Ax_k = b - AQ_k y_k = b - Q_{k+1}\tilde{T}_k y_k$$

- Let $Q_k^T r_k = Q_k^T \left(b - Q_{k+1}\tilde{T}_k y_k\right) = 0$ (i.e., $r_k \perp \mathcal{K}_k$), we obtain

$$Q_k^T Q_{k+1}\tilde{T}_k y_k = Q_k^T b$$

  where $Q_k^T Q_{k+1}\tilde{T}_k = T_k$ and $Q_k^T b = \beta e_1$ with $\beta = \|b\|$

- Hence,

$$T_k y_k = \beta e_1 \tag{2}$$

  where $T_k = Q_k^T A Q_k$ is tridiagonal, and is SPD if A is SPD

- It takes $\mathcal{O}(1)$ flops to update Cholesky factorization of T_k and then $\mathcal{O}(k)$ flops to solve (2). Resulting algorithm is equivalent to CG

Outline

1 CG as Optimization Method

2 CG and Krylov Subspace

3 Convergence Properties of CG

Termination in Exact Arithmetic

Theorem (Theorem 11.3.1 in MC p. 629)

If k_* is dimension of smallest invariant space that contains r_0, then CG terminates in k_* steps in exact arithmetic.

- A subspace $\mathcal{S}$ is *invariant* w.r.t. to A if for any $v \in \mathcal{S}$, $Av \in \mathcal{S}$

Proof.

$r_0 = b$ can be written as a linear combination of k_* eigenvectors of A, $\{v_1, v_2, \ldots, v_{k_*}\}$, so is $x_* = A^{-1}b$ (since A is diagonalizable).
At step $k = k_*$, $\dim(\mathcal{K}_{k_*}) = k_*$, and $\{v_1, v_2, \ldots, v_{k_*}\}$ form a basis of $\mathcal{K}_{k_*}$, and hence $x_* \in \mathcal{K}_{k_*}$.
If $x_* \in \mathcal{K}_k$ for $k < k_*$, $\dim(\mathcal{K}_k) = k < k_*$, then r_0 would have been contained in a lower-dimensional invariant space. Contradiction. □

- If A has s distinct eigenvalues, CG converges in $\leq s$ iterations.
- With rounding errors, we may not get exact x_* after k_* iterations
- In addition, we may want to terminate sooner than k_* iterations

Optimality of Conjugate Gradients

Theorem (Theorem 38.2 in NLA p. 296)

If $r_{k-1} \neq 0$, then error $e_k = x_* - x_k$ is minimized in A-norm in $\mathcal{K}_k$.

Proof.

Consider arbitrary point $x = x_k - \Delta x \in \mathcal{K}_k$ with error $e = x_* - x = e_k + \Delta x$. So

$$\begin{aligned}\|e\|_A^2 &= (e_k + \Delta x)^T A (e_k + \Delta x) \\ &= e_k^T A e_k + \Delta x^T A \Delta x + 2 e_k^T A \Delta x,\end{aligned}$$

where $e_k^T A \Delta x = r_k^T \Delta x = 0$ because $r_k \perp \mathcal{K}_k$. Since A is SPD, $\|e\|_A^2 \geq \|e_k\|_A^2$ and equality holds iff $\Delta x = 0$. □

- Because $\mathcal{K}_k$ grows monotonically, $\|e_k\|_A$ decreases monotonically
- Note: A-norm is defined as $\|x\|_A = \sqrt{x^T A x}$, assuming A is SPD. It is different from weighted norm $\|x\|_W = \|Wx\|$

Convergence Rate with Rounding Errors

- If A has 2-norm condition number κ, error is bounded by

$$\frac{\|e_k\|_A}{\|e_0\|_A} \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k$$

- Proof is based on analysis of matrix polynomials
    - CG minimizes $\|p_k(A)e_0\|_A$ at kth step, with $e_0 = x_*$, where p_k is degree-k polynomial $p_k(x) = 1 + c_1 x + c_2 x^2 + \cdots + c_k x^k$
    - $\|e_k\|_A / \|e_0\|_A \leq \inf_{p_k} \max_\lambda |p_k(\lambda)|$, where λ are eigenvalues of A, which is further bounded using theory of orthogonal polynomials
- $2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \approx 2\left(1 - \frac{2}{\sqrt{\kappa}}\right)^k$ for large κ, so CG takes up to $O(\sqrt{\kappa})$ iterations
- In general, CG performs well with clustered eigenvalues