AMS526: Numerical Analysis I
(Numerical Linear Algebra)
Lecture 25: More on Preconditioners;
Overview of Multigrid Methods

Xiangmin Jiao

SUNY Stony Brook
Outline

1. More on Preconditioners

2. Smoothing Effect of Stationary Iterative Methods

3. Multigrid Method
   - Motivation
   - Key Ideas of Multigrid
   - More Advanced Topics
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?

Suppose $M$ is Hermitian positive definite, with $M = CC^*$ for some $C$, then $Ax = b$ is equivalent to $\left[ C^{-1}AC^{-*} \right] \left( C^*x \right) = C^{-1}b$, 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 $M^{-1}A$.
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

  $$\left[C^{-1}AC^{-*}\right] (C^*x) = C^{-1}b,$$

  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

\[
\begin{align*}
x_0 &= 0, \quad r_0 = b, \quad p_0 = M^{-1}r_0, \quad z_0 = p_0 \\
\text{for } n &= 1, 2, 3, \ldots \\
\alpha_n &= (r_{n-1}^T z_{n-1})/(p_{n-1}^T Ap_{n-1}) \quad \text{step length} \\
x_n &= x_{n-1} + \alpha_n p_{n-1} \quad \text{approximate solution} \\
r_n &= r_{n-1} - \alpha_n Ap_{n-1} \quad \text{residual} \\
z_n &= M^{-1}r_n \quad \text{preconditioning} \\
\beta_n &= (r_n^T z_n)/(r_{n-1}^T z_{n-1}) \quad \text{improvement this step} \\
p_n &= z_n + \beta_n p_{n-1} \quad \text{search direction}
\end{align*}
\]
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.
Outline

1. More on Preconditioners

2. Smoothing Effect of Stationary Iterative Methods

3. Multigrid Method
   - Motivation
   - Key Ideas of Multigrid
   - More Advanced Topics
Stationary Iterative Methods

- Stationary iterative methods can be interpreted as a fixed point iteration obtained by matrix splitting.
- Let $A = M - N$ and $r_k = b - Ax_k$ we can obtain

$$x_{k+1} = M^{-1}Nx_k + M^{-1}b$$  \hspace{1cm} (1)

$$x_{k+1} = x_k + M^{-1}r_k$$  \hspace{1cm} (2)

- Different choices of splitting lead to various schemes
Stationary Iterative Methods

- These iteration schemes work for a wide range of problems
- They can often be implemented without forming the matrix explicitly.
- However, they have slow convergence

Example

For 2D Poisson equation,
Spectral radius of Jacobi iteration matrix is \( \cos \left( \frac{\pi}{n} \right) \approx 1 - O \left( \frac{1}{n^2} \right) \).
Number of iterations required to achieve \( \epsilon \) is \( O(n^2 \ln \epsilon^{-1}) \).
After 5 Jacobi iterations on a Poisson equation, error decreases very slowly.
Smoothing Effect

- The reason behind this behavior is the smoothing property of stationary iterative methods.
- This property is one of the theoretical foundations of multigrid methods.
- To illustrate the idea we apply iterative methods to the homogeneous system with initial guess $v_k$

$$Au = 0$$

- $v_k$ is chosen as $(v_k)_j = \sin \left( \frac{j k \pi}{n} \right), \; 1 \leq k \leq n - 1, \; 1 \leq j \leq n - 1$ (Fourier modes)
Smoothing Effect

- The modes in the lower half of the spectrum, with wavenumbers in the range $1 \leq k < \frac{n}{2}$ are called *low frequency* or smooth modes.
- The modes in the upper half of the spectrum, with $\frac{n}{2} \leq k \leq n - 1$ are called *high frequency* modes or *oscillatory* modes.

![Diagram of modes with wavenumbers k = 1, 3, 6]
Smoothing Effect

Weighted Jacobi applied on 1-D model problem with 64 points with initial guess \( v_1, v_3 \) and \( v_6 \)
Gauss Seidel applied on 1-D model problem with 64 points with initial guess $v_1, v_3$ and $v_6$
Smoothing Effect

- Oscillatory modes are eliminated quickly
- Smooth modes remain relatively unchanged
- Errors for the model problem can be decomposed using these Fourier modes
- After several iterations, high frequency components will disappear and the error becomes smooth
Smoothing Effect

If we project a smooth wave directly onto a coarser grid, it becomes more oscillatory.
Smoothing Effect

- What does it imply?
- If we can move the error to a coarser grid, iterations will be more effective!
- Even if the error does not become more oscillatory, relaxing on the coarse grid is simply cheaper
- We may consider using coarse grids
Outline

1. More on Preconditioners

2. Smoothing Effect of Stationary Iterative Methods

3. Multigrid Method
   - Motivation
   - Key Ideas of Multigrid
   - More Advanced Topics
We can solve problems on coarse grids to obtain better initial guesses:

- Relax $Au = f$ on a very coarse grid $\Omega^{8h}$ to obtain an initial guess for the next finer grid $\Omega^{4h}$.
- Relax $Au = f$ on grid $\Omega^{4h}$ to obtain an initial guess for $\Omega^{2h}$.
- Relax $Au = f$ on grid $\Omega^{2h}$ to obtain an initial guess for $\Omega^h$.
- Relax $Au = f$ on $\Omega^h$ to obtain a final approximation to the solution.
Correction Scheme

- From our previous observations, error becomes smooth after relaxations.
- If we move the error to a coarser grid, it becomes oscillatory and iterations are effective.
- What problem should we be solving then?
- The residual equation \( Ae = f - Av = r \), where \( v \) is approximate solution of \( u \).
Correction Scheme

- Why residual equation?
- We want to relax the error directly since it becomes oscillatory on coarse level
- If we can solve the residual equation accurately then the real solution \( u \) can be obtained by \( u = v + e \).
- Relaxation on the original equation \( Au = f \) with arbitrary initial guess \( v \) is equivalent to relaxing on the residual equation \( Ae = r \) with specific initial guess \( e = 0 \).
Two-Grid Correction Scheme

The basic form of the multigrid method is defined as the following two-grid correction scheme [Briggs et al., Multigrid Tutorial]:

\[ v^h \leftarrow MG(v^h, f^h) \]

1. **Presmoothing:** relax \( \mu_1 \) times on \( A^h u^h = f^h \) on \( \Omega^h \) with initial guess \( v^h \).

2. **Restriction:** compute the fine-grid residual \( r^h = f^h - A^h v^h \) and restrict it to the coarse grid by \( r^{2h} = R r^h \).

3. **Coarse Grid Solving:** either solve \( A^{2h} e^{2h} = r^{2h} \) or relax \( \mu_3 \) times with initial guess 0 on \( \Omega^{2h} \).

4. **Prolongation:** interpolate the coarse-grid error to the fine grid by \( e^h = Pe^{2h} \) and correct the fine-grid approximation by \( v^h \leftarrow v^h + e^h \).

5. **Postsmoothing:** Relax \( \mu_2 \) times on \( A^h u^h = f^h \) on \( \Omega^h \) with initial guess \( v^h \).
Presmoothing: relax $\mu_1$ times on $A^h u^h = f^h$ on $\Omega^h$ with initial guess $v^h$.

- We apply $\mu_1$ steps of iterations on the original linear system.
- This step is known as the presmoothing step.
- After iterations errors $e^h$ will become smooth and it will appear oscillatory on $\Omega^{2h}$.
- We then approximate the residual equation on $\Omega^{2h}$. 
Restriction

Restriction: compute the fine-grid residual $r^h = f^h - A^h v^h$ and restrict it to the coarse grid by $r^{2h} = R r^h$.

- We restrict the residual onto $\Omega^{2h}$
- Restriction operator can be chosen as injection $v_j^{2h} = v_{2j}^h$ or full-weighting $v_j^{2h} = \frac{1}{4}(v_{2j-1}^h + 2v_{2j}^h + v_{2j+1}^h)$

Figure: Restriction by full weighting
Solving on coarse level

**Coarse Grid Solving**: either solve \( A^{2h} e^{2h} = r^{2h} \) or relax \( \mu_3 \) times with initial guess 0 on \( \Omega^{2h} \).

- We obtain \( A^{2h} \) by rediscretizing the PDE on \( \Omega^{2h} \)
- \( A^{2h} e^{2h} = r^{2h} \) is an approximation of \( A^h e^h = r^h \) on \( \Omega^{2h} \)
- Iterative methods are effective as \( e^{2h} \) becomes oscillatory
- Iterations are also cheaper as there are less grid points
**Prolongation**

**Prolongation**: interpolate the coarse-grid error to the fine grid by 
\[ e^h = P e^{2h} \] and correct the fine-grid approximation by 
\[ v^h \leftarrow v^h + e^h. \]

- After \( e^{2h} \) is obtained, we interpolate it back to \( \Omega^h \) and update error.
- Prolongation operator can be chosen as linear interpolation
  \[ v_{2j}^h = v_j^{2h}, \quad v_{2j+1}^h = \frac{1}{2}(v_j^{2h} + v_{j+1}^{2h}) \]

![Figure: Prolongation by linear interpolation](image)
Postsmoothing

**Postsmoothing**: relax $\mu_2$ times on $A^h u^h = f^h$ on $\Omega^h$ with initial guess $v^h$.

- We apply $\mu_2$ steps of iterations on the original linear system
- This step is known as the postsmoothing step
- Errors will be further reduced
Two-Grid Correction Scheme

- Iteration on fine grid leaves smooth errors $e^h$ and they appear to be oscillatory on coarse grid as $e^{2h}$.
- Iteration on coarse grid then solve $e^{2h}$ effectively and $e^{2h}$ will become a good approximation of $e^h$ after interpolation.
- Finally with the correction step $v^h \leftarrow v^h + e^h$, we will obtain a solution very close to $u$.
- Errors which cannot be eliminated effectively by iterations are removed by coarse grid correction.
Toward Multigrids

- In our description, we assume $e^{2h}$ on coarse level is solved accurately.
- Practically a few steps of iterations cannot guarantee sufficient accuracy of $e^{2h}$.
- We may apply two-grid idea recursively on subsequent levels.
- We can recursively solve problems on coarser levels and use them as initial guesses on fine levels.
V-cycle and Full Multigrid scheme

Figure: V-cycle and FMG scheme
Smooth errors are defined algebraically as $e_{k+1} \approx e_k$ which leads us to $Ae \approx 0$

Define interpolation operator $P$

Coarsening is performed by a greedy maximum independent set algorithm on weighted graph

Restriction is chosen as the transpose of interpolation and $A_{2h} = P^T A_h P$

Current AMG development focuses on improving coarsening strategy and interpolation formula
Numerical Results: Poisson Equation

- Poisson equation discretized using third-order generalized finite difference method. The resulting matrix is asymmetric
- Comparison of algebraic multigrid method, MATLAB’s built in direct solver and GMRES to problems of various sizes

<table>
<thead>
<tr>
<th>size</th>
<th>AMG ($TOL = 10^{-10}$)</th>
<th>Direct Solver</th>
<th>GMRES(10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 × 10 × 10</td>
<td>0.0081 seconds</td>
<td>0.0093 seconds</td>
<td>0.020 seconds</td>
</tr>
<tr>
<td>20 × 20 × 20</td>
<td>0.081 seconds</td>
<td>0.21 seconds</td>
<td>0.28 seconds</td>
</tr>
<tr>
<td>40 × 40 × 40</td>
<td>0.95 seconds</td>
<td>8.14 seconds</td>
<td>7.15 seconds</td>
</tr>
<tr>
<td>80 × 80 × 80</td>
<td>11.79 seconds</td>
<td>out of memory</td>
<td>470.88 seconds</td>
</tr>
</tbody>
</table>