Algebraic Multigrid as Solvers and as Preconditioner

Domenico Lahaye
domenico.lahaye@cs.kuleuven.ac.be
http://www.cs.kuleuven.ac.be/~domenico/

Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200 A
B-3001 Heverlee
Belgium
Structure of the Presentation

- Motivation for Iterative solvers
- Stationary Iterative Solvers
- Multigrid Methods
- Algebraic Multigrid Methods
- Krylov Subspace Methods
- Algebraic Multigrid Methods as Preconditioner
Motivation for Iterative solvers

- 2D Elliptic partial differential equation

\[- \frac{\partial}{\partial x} \left( \nu \frac{\partial u}{\partial x} \right) - \frac{\partial}{\partial y} \left( \nu \frac{\partial u}{\partial y} \right) = f \]

- Discretization by e.g. finite difference or finite elements

- Resulting system of linear equation \( Ax = b \)
  * \( A \) discrete differential operator
  * \( x \) and \( b \) unknown function \( u \) and \( f \) in discrete points respectively

- \( \text{dim}(A) \) large (accurate discretizations)

- \( Ax = b \) has to be solved repeatedly (parameter studies)

  \( \Rightarrow \) Fast and memory efficient solution procedures required
Motivation for Iterative solvers (2)

Example: Numerical models for electrical motors

\[ \mathbf{A} = (0, 0, u) \quad \mathbf{B} = \nabla \times \mathbf{A} \quad \mathbf{H} = \nu \mathbf{B} \]

\( \Rightarrow \) computation of forces and torques

\[ \mathbf{A} x = b \quad \text{dim}(\mathbf{A}) \sim 10^5 - 10^6 \]

\( \Rightarrow \) solved in time stepping or optimization loop
Iterative methods

Given $x_0$, compute $\{x_m\}_{m \geq 1}$ such that

$\| e_m \| = \| x - x_m \| \to 0$ if $m \to \infty$

- **stationary methods**
  - multigrid methods
  - domain decomposition methods
- **non-stationary methods**
  - Krylov subspace methods

Matrix-vector multiply as computational kernel

$\Rightarrow$ sparsity exploited

$\Rightarrow$ $A v$ in $O(N)$ flops, $N = \dim(A)$

Direct methods

Rely on a factorization of $A$
Representative sparsity structure of $A$
Structure of the Presentation

- Motivation for Iterative solvers
- Stationary Iterative Solvers
- Multigrid Methods
- Algebraic Multigrid Methods
- Krylov Subspace Methods
- Algebraic Multigrid Methods as Preconditioner
- Conclusions
Basic Iterative Schemes for $Ax = b$

- **Matrix splitting** $A = M - N$ with $M^{-1}v$ “easy” to compute

- **Iterative scheme**
  
  $x^{m+1} = M^{-1} N x^m + M^{-1} b$  
  (iterand)  
  $e^{m+1} = (I - M^{-1} A) e^m$  
  (error)

- **$A = D - L - U$**
  
  * Jacobi method $M_{JAC} = D$  
  (diagonal)
  
  * Gauss-Seidel method $M_{GS} = D - L$  
  (triangular)
  
  * Successive relaxation $M_{SOR} = \frac{1}{\omega} D - L$  
  $\omega > 0$

- **Asymptotic** rate of convergence given by $\rho(I - M^{-1} A)$ or $\| I - M^{-1} A \|$
Basic Iterative Schemes for $A x = b$ (2)

Model Problem Analysis

- Continuous problem: $-\Delta u = f$ on $\Omega = [0, 1] \times [0, 1] + \text{boundary conditions}$

- Discretization on uniform mesh, mesh width $h$, using central finite differences

- Linear system $A x = b$ with $[A] = \frac{1}{h^2} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$  
  \[ \Omega^h = \]

Domenico Lahaye
Basic Iterative Schemes for $Ax = b$ (3)

Convergence history of lexicographic Gauss-Seidel (fixed mesh)

* for a fixed mesh (fixed $h$ )
  initial stage: fast convergence
  later stage: stalling convergence

\[
\begin{align*}
10^1 & \quad 10^2 \quad 10^3 \quad 10^4 \\
0 & \quad 10 & \quad 20 & \quad 30 & \quad 40 & \quad 50 & \quad 60
\end{align*}
\]
Basic Iterative Schemes for $Ax = b$ (4)

Convergence history of lexicographic Gauss-Seidel (varying mesh size)

* for finer meshes ($h \to 0$)

$\| I - M^{-1} A \| = 1 - O(h^2)$

$\Rightarrow$ increasing number of iterations
Basic Iterative Schemes for $Ax = b$ (5)

Slow convergence is caused by smooth error components

Initial error  After 5 iterations  After 10 iterations

Smooth error components can be represented on coarser grids
Motivation for Iterative solvers
Stationary Iterative Solvers
Multigrid Methods
Algebraic Multigrid Methods
Krylov Subspace Methods
Algebraic Multigrid Methods as Preconditioner
Conclusions
Multigrid Methods

- exploit PDE background of the linear problem
- multigrid methods = smoother (basic iterative scheme) + coarse grid correction
- two-grid scheme

multigrid scheme by recursive application to solve coarse grid problem
Multigrid Methods (2)

Geometric multigrid methods

\[ A^h x^h = b^h \text{ on } \Omega^h \implies A^H x^H = b^H \text{ on } \Omega^H \]

\[ \text{construct restriction } I^H_h : \Omega^h \rightarrow \Omega^H \quad \text{interpolation } I^h_H : \Omega^H \rightarrow \Omega^h \]

\[ \text{coarse grid correction} \quad K_{h,H} = I^h - I^h_H (A^H)^{-1} I^H_h A^h \]

\[ \text{smoother} \quad S^h = (I - (Q^h)^{-1} A^h) \quad Q^h \text{ splitting of } A^h \]

\[ \text{multigrid iteration matrix} \quad M_{h,H}(\nu_1, \nu_2) = (S^h_2)^{\nu_2} K_{h,H} (S^h_1)^{\nu_1} \]
Multigrid Methods (3)

Geometric multigrid methods (2)

- Different orders in visiting coarser grids ⇒ different cycles (V-, W-, F-cycle)

- $M_{h,H}$ multigrid iteration matrix
  
  $\| I - (M_{h,H})^{-1} A \| = C$
  
  $C$ small independent of $h$
  
  ⇒ mesh-width independent ($h$-independent) convergence
Multigrid Methods (4)

- Multigrid methods require a hierarchy of grids.
- This hierarchy can be cumbersome to construct.
- Example left: $\Omega^H = ???$
Structure of the Presentation

- Motivation for Iterative solvers
- Stationary Iterative Solvers
- Multigrid Methods
  - Algebraic Multigrid Methods
- Krylov Subspace Methods
- Algebraic Multigrid Methods as Preconditioner
- Conclusions
Algebraic Multigrid Methods

- Automatic construction of $\Omega^H$ and $A^H x^H = b^H$

- Using information contained in $A^h$ only and no information on
  - differential equation
  - geometry of $\Omega^h$

- notion of strength of coupling between grid points exploited

- Strength of coupling influences the smoother

- Illustration for
  - anisotropic diffusion equation
  - diffusion equation with discontinuous diffusion coefficient
Anisotropic Diffusion Equation

- Equation \(-\epsilon \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = f\) with \(\epsilon \ll 1\) on \([0, 1] \times [0, 1]\)

- Finite difference discretization on \(\Omega^h\) \(A^h x^h = b^h\) with

\[
[A] = \frac{1}{h^2} \begin{bmatrix}
0 & -1 & 0 \\
-\epsilon & 4 & -\epsilon \\
0 & -1 & 0
\end{bmatrix}
\]

- weak coupling in \(x\)-direction, strong coupling in \(y\)-direction

- after point Gauss-Seidel smoothing, error is *smooth* in \(y\)-direction
  *oscillatory* in \(x\)-direction

- coarsening feasible in \(y\)-direction only!
Algebraic Multigrid Methods (3)

Initial error

After 5 iterations

After 10 iterations

\[ \Omega^h = \begin{array}{|c|c|c|c|} \hline \end{array} \quad \Rightarrow \quad \Omega^H = \begin{array}{|c|c|c|c|} \hline \end{array} \]
Problem with discontinuous diffusion coefficient

\[ -\frac{\partial}{\partial x} \left( \nu \frac{\partial u}{\partial x} \right) - \frac{\partial}{\partial y} \left( \nu \frac{\partial u}{\partial y} \right) = f \quad \text{with} \quad \nu = \begin{cases} 0.0 \\ 0.375 \\ 0.625 \\ 1.0 \end{cases} \]

\( \nu = 1 \)
\( \nu = 10^6 \)

\( x \)
\( y \)

\( 0.0 \)
\( 0.375 \)
\( 0.625 \)
\( 1.0 \)

- after point Gauss-Seidel smoothing, error oscillatory across interface of discontinuity
- coarsening not feasible “close to” the interface
Algebraic Multigrid Methods (5)

Initial error

After 5 iterations

After 10 iterations

\[ \Omega^h = \begin{array}{cccc} & & & \\ & & & \\ & & & \\ & & & \end{array} \quad \Rightarrow \quad \Omega^H = \begin{array}{cccc} & & & \\ & & & \\ & & & \end{array} \]
Algebraic Multigrid Methods (6)

- **geometric** multigrid
  - construct hierarchy of coarser grids
  - smoother adapted to constructed hierarchy
    - ⇒ work invested in the smoother
    - ⇒ e.g. using line smoothers

- **algebraic** multigrid
  - simple, fixed smoother: point Gauss-Seidel
  - coarse grid correction adapted to the local properties of the smoother
    - ⇒ work invested in the coarse grid correction
Setup phase

- construction of splitting $\Omega^h = C^h \cup F^h$ and set $\Omega^H = C^h$
- construction of interpolation $I^h_H$
- strength of coupling between nodes coded in $A^h$ exploited
- matrix dependent interpolation: $(I^H_h)_{ij} \sim A^h_{ij}/A^h_{ii}$
- Galerkin coarsening: $A^H = I^H_h A^h I^h_H$
- apply recursively using $A^H$ as input

Solve phase

- multigrid cycling
Algebraic Multigrid Methods (8)

Permanent Magnet Machine Test Case
Adaptive construction of triangulation in 13 refinement steps
Initial and intermediate triangulation
Number of iterations for AMG

* multigrid behavior
CPU-time measurements for AMG

- RAMG speedup by factor 7.5
Structure of the Presentation

- Motivation for Iterative solvers
- Stationary Iterative Solvers
- Multigrid Methods
- Algebraic Multigrid Methods
- Krylov Subspace Methods
- Algebraic Multigrid Methods as Preconditioner
- Conclusions
Non-stationary iterative methods for solving $A x = b$

Given $x_0$ and $r_0 = b - A x_0$
find $x_m \in x_0 + K^m(A, r_0) = \text{span}\{r_0, A r_0, \ldots, A^{m-1} r_0\}$

Given $V_m = [v_1, \ldots, v_m]$ a basis for $K^m(A, r_0)$ \Rightarrow $x_m = x_0 + V_m y_m$

$y_m$ a vector of $m$ coefficients

basis $V_m$ * Arnoldi method \hspace{1cm} \text{vector } y_m$ * residual projection methods
* bi-Lanczos method \hspace{1cm} * residual norm minimization
* Lanczos method

No Krylov method optimal for large class of problems
The Arnoldi Method for constructing $V_m$

- Initial vector $v_1 = r_0 / \| r_0 \|$

- Given $V_m = [v_1, \ldots, v_m]$ a basis for $K^m(A, r_0)$
  compute $A v_m$ and orthogonalize with respect to $V_m$
  normalize and set resulting vector equal to new basis vector $v_{m+1}$

- recurrence relation $A V_m = V_{m+1} \tilde{H}_{m+1,m}$
  $\tilde{H}_{m+1,m} = \begin{pmatrix} H_{m,m} \\ 0 \ldots 0 h_{m+1,m} \end{pmatrix}$
  $H_{m,m}$ upper Hessenberg

- long term recurrences: expensive in memory and CPU-time
The Bi-Orthogonal Lanczos Method for constructing $V_m$

- Avoid long term recurrences using bi-orthogonality

$$V_m^T W_m = D_m = \text{diag}[d_1, \ldots, d_m] \quad W_m \text{ basis for } K^m(A^T, r_0^*)$$

- Recurrence relation $A V_m = V_{m+1} \tilde{T}_{m+1}$

$$\tilde{T}_{m+1} \text{ tridiagonal } \Rightarrow \text{ short term recurrences}$$

- (Near) breakdown if $v_i^T w_i = 0 \ (v_i^T w_i \sim 0)$
Krylov Subspace Methods (4)

Determining the vector $y_m$

$x_m \in x_0 + K^m(A, r_0) \Rightarrow r_m = b - A x_m = r_0 - A V_m y_m$

Residual projection methods $\Leftrightarrow V_m^T r_m = 0$
$\Leftrightarrow H_{m,m} y_m = \beta_1 e_1$
$m \times m$ linear system: solve by LU-factorization

Residual minimization methods $\Leftrightarrow \text{minimize } \| r_m \|$  
$\Leftrightarrow H_{m+1,m} y_m = \beta_1 e_1$
$(m + 1) \times m$ linear system: solve in least-squares sense
\[ Ax = b \text{ with } A \text{ symmetric, positive definite} \]

\[ A = A^T \Rightarrow H_{m,m} = (H_{m,m})^T \Rightarrow H_{m,m} \text{ tridiagonal} \]

\[ A \text{ pos. def.} \Rightarrow H_{m,m} = V_m^T A V_m \text{ pos. def.} \Rightarrow \text{LU-fact. without pivoting} \]

\[ \text{update LU-fact. from step } m \text{ to } m + 1 \]

\[ \text{coupled 2-term recurrences} \quad \{r_{m-1}, p_{m-1}, r_m, p_m\} \Rightarrow \{r_{m+1}, p_{m+1}\} \]

\[ p_i^T A p_j = 0 \text{ for } i \neq j \Rightarrow \text{conjugate gradient method (CG)} \]
Krylov Subspace Methods (6)

Convergence of Krylov Subspace Methods

- Krylov subspace methods converge superlinearly

- Bound for CG

\[ \| x - x_m \|_A \leq 2 \left( \frac{\sqrt{\text{cond}_2(A)} - 1}{\sqrt{\text{cond}_2(A)} + 1} \right)^m \| x - x_0 \|_A \]

- Preconditioning: \( A x = b \Rightarrow M^{-1} A x = M^{-1} b \)

  * \( M^{-1} A \approx I \)
  * \( M z = r \) “easy” to solve

- Preconditioned CG: \( \forall m \) replace \( r_m \) by \( z_m = M^{-1} r_m \)

- Possible preconditioners for CG: \( M_{JAC}, M_{SSOR}, \ldots \)
The Convergence of CG: A closer look

- CG: \( \forall m \quad T_m y_m = \beta_1 e_1 \quad T_m = V_m^T A V_m \)

- Eigenvalues of \( T_m \) are called Ritz-values

- Ritz-values converge to \( \text{spec}(A_m) \) for increasing \( m \)
  fast converges towards “outliers” in \( \text{spec}(A_m) \), if any
cfr. Krylov subspace methods for eigenvalue computations

- Convergence of Ritz values governs convergence of CG
Structure of the Presentation

- Motivation for Iterative solvers
- Stationary Iterative Solvers
- Multigrid Methods
- Algebraic Multigrid Methods
- Krylov Subspace Methods
- Algebraic Multigrid Methods as Preconditioner
- Conclusions
AMG as Preconditioner

Problem with anti-periodic boundary conditions

\[ A_z|_{\Gamma_1} = -A_z|_{\Gamma_2} \]

<table>
<thead>
<tr>
<th>Adaptive refinement step</th>
<th>Number of iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>classical AMG</td>
</tr>
<tr>
<td>0</td>
<td>148</td>
</tr>
<tr>
<td>1</td>
<td>div</td>
</tr>
<tr>
<td>2</td>
<td>div</td>
</tr>
<tr>
<td>3</td>
<td>div</td>
</tr>
<tr>
<td>4</td>
<td>div</td>
</tr>
<tr>
<td>5</td>
<td>div</td>
</tr>
<tr>
<td>6</td>
<td>div</td>
</tr>
</tbody>
</table>
convergence of the Ritz values towards $\text{spec}(I - M^{-1}A)$ during CG-iteration

spec($I - M^{-1}A$) has a few outliers

outliers are captured fast by CG-iteration
AMG as Preconditioner (3)

Permanent Magnet Machine Test Case
AMG as Preconditioner (4)

Number of iterations for AMG

- CG accelerates and stabilizes convergence
CPU-time measurements for AMG

* AMG speedup by factor 7.5
* AMG/CG speedup by factor 15
Conclusions

- We presented multigrid and algebraic multigrid for two-dimensional diffusion problems.

- Algebraic multigrid methods allow to solve problems on complicated geometries.

- Algebraic multigrid methods are more efficient than single-level iterative solvers.

- Using algebraic multigrid as a preconditioner improves its stability and speed.