Introduction to Multilevel Solvers for the Physical Sciences

Chris Siefert

Scalable Algorithms Group

Sandia National Laboratories
SAND 2007-7657P
Outline

- Background.
- Computation in the Physical Sciences.
- Solving Linear Systems with Iterative Methods.
- Introduction to Multilevel Methods.
- Open Questions in Multilevel Methods.
About Me

- **B.S. W&M ’00**
  - Double Major (CS & Math).
  - Research in optimization & applied statistics w/ Torczon and Trosset (Indiana).

- **Ph.D. UIUC ’06**
  - CS w/ Computational Science & Engineering option.
  - Research in numerical linear algebra w/ de Sturler(VT).

- Sandia National Laboratories, Postdoc
  - Scalable algorithms group.
  - Research in multilevel methods w/ Tuminaro and Hu.
Course Background

- Assumed audience background:
  - Multivariable calculus (MATH 212).
  - Linear algebra (MATH 211).

- A more detailed talk would require:
  - Algorithms (CS 303).
  - Advanced linear algebra (MATH 408).
  - Numerical analysis (MATH 413, 414).
Outline

- Background.
- Computation in the Physical Sciences.
- Solving Linear Systems with Iterative Methods.
- Introduction to Multilevel Methods.
- Open Questions in Multilevel Methods.
What is Computational Science?

- What do we think of when we think of computational science?
  - Usually “big” things. . .
  - Airplanes, cars, rockets, etc.
What is Computational Science?

What do we think of when we think of computational science?
- Usually “big” things...
- Airplanes, cars, rockets, etc.

BUT computational science touches everyday things as well!
Process of Computational Science

- Model the problem.
- Discretize the model.
- Solve the discrete problem.
- Analyze results.
Process of Computational Science

- Model the problem.
- Discretize the model.
- Solve the discrete problem.
- Analyze results.

Note: There are more “steps,” which I am neglecting.
Model the Problem

“All models are wrong; some models are useful” – George Box

- For this talk, we consider only PDE-based models.
- Example problem: thermal diffusion on a beam.

- Model: Heat Equation

\[ \frac{\partial u}{\partial t} = c \frac{\partial^2 u}{\partial x^2} \]
Discretize the Problem

“Truth is much too complicated to allow anything but approximations” – John von Neumann

Problem must be discrete to solve on a computer.

Why not analytic methods?
  - Complicated geometries.
  - Complicated physics.
  - Solution may not exist.

Analytic methods critical for verification & validation.

Types of discretization: Finite difference, finite element, finite volume.
Example: Finite Differences (1)

- Limit definition of derivative:

\[ f'(x) = \lim_{h \to 0} \frac{f(x + h) - f(x)}{h} \]

- Basic idea: pick a finite \( h \).

\[ f'(x) \approx \frac{f(x + h) - f(x)}{h} \]

- We can do this for 2nd derivatives as well:

\[ f''(x) \approx \frac{f(x + h) - 2f(x) + f(x - h)}{h^2} \]
Example: Finite Differences (2)

- Model:

\[ \frac{\partial u}{\partial t} = c \frac{\partial^2 u}{\partial x^2} \]

- Discretization (subscript = space, superscript = time):

\[ \frac{U_{j}^{k+1} - U_{j}^{k}}{\Delta t} = c \frac{U_{j+1}^{k+1} - 2U_{j}^{k+1} + U_{j-1}^{k+1}}{(\Delta x)^2} \]

- Mesh:

\[ \Delta x \quad \Delta x \quad \Delta x \quad \Delta x \]
Solve the Discrete Problem

“Mathematics is the queen of the sciences” – Carl Friedrich Gauss

- Discretization (subscript = space, superscript = time):

\[
\frac{U_j^{k+1} - U_j^k}{\Delta t} = c \frac{U_{j+1}^{k+1} - 2U_j^{k+1} + U_{j-1}^{k+1}}{(\Delta x)^2}
\]

- This is a linear system:

\[
\begin{bmatrix}
-\frac{c}{(\Delta x)^2} & \left(2\frac{c}{(\Delta x)^2} + \frac{1}{\Delta t}\right) & -\frac{c}{(\Delta x)^2}
\end{bmatrix}
\begin{bmatrix}
U_{j-1}^{k+1} \\
U_j^{k+1} \\
U_{j+1}^{k+1}
\end{bmatrix}
= \frac{U_j^k}{\Delta t}
\]

for \( j = 1, \ldots, n \).
Analyze the Results

“When you are solving a problem, don’t worry. Now, after you have solved the problem, then that’s the time to worry.” – Richard Feynman

- Is there something we missed in the model?
- Does the answer look plausible?
- Does the answer match experiment (if applicable)?
- Does the answer converge with mesh refinement?
- What does the answer tell us about the underlying problem?
Outline

- Background.
- Computation in the Physical Sciences.
- **Solving Linear Systems with Iterative Methods.**
- Introduction to Multilevel Methods.
- Open Questions in Multilevel Methods.
Importance of Linear Algebra

- Solving linear systems was critical to the example
  ⇒ One linear solve per time step!
- This is true of many simulations.
- We can do this w/ Gaussian elimination (GE).
- But is it fast enough?
- How long does GE take for an $n \times n$ matrix?
- We need time complexity analysis!
Gaussian Elimination

\[
\begin{bmatrix}
  a & b & c \\
  d & e & f \\
  g & h & i \\
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  z \\
\end{bmatrix}
=
\begin{bmatrix}
  \alpha \\
  \beta \\
  \gamma \\
\end{bmatrix}
\]

Total Operations = 0
Gaussian Elimination

\[
\begin{bmatrix}
1 & b & c \\
d & e & f \\
g & h & i \\
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
\end{bmatrix}
= \begin{bmatrix}
\alpha \\
\beta \\
\gamma \\
\end{bmatrix}
\]

Total Operations \( \approx n \)

1. Divide through the 1st row by \( a \).
Gaussian Elimination

\[
\begin{bmatrix}
1 & b & c \\
0 & e & f \\
g & h & i
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix} =
\begin{bmatrix}
\alpha \\
\beta \\
\gamma
\end{bmatrix}
\]

Total Operations \( \approx 2n \)

1. Divide through the 1st row by \( a \).
2. Subtract off \( d \) times the first row from the second.
Gaussian Elimination

\[
\begin{bmatrix}
1 & b & c \\
0 & e & f \\
0 & h & i \\
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
\end{bmatrix}
= 
\begin{bmatrix}
\alpha \\
\beta \\
\gamma \\
\end{bmatrix}
\]

Total Operations \( \approx n^2 \)
1. Divide through the 1st row by \( a \).
2. Subtract off \( d \) times the first row from the second.
3. Do the same for the remaining \( n - 2 \) rows.
Gaussian Elimination

\[
\begin{bmatrix}
1 & b & c \\
0 & 1 & f \\
0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z \\
\end{bmatrix}
= 
\begin{bmatrix}
\alpha \\
\beta \\
\gamma \\
\end{bmatrix}
\]

Total Operations \(\approx n^3\)
1. Divide through the 1st row by \(a\).
2. Subtract off \(d\) times the first row from the second.
3. Do the same for the remaining \(n - 2\) rows.
4. Repeat the for the remaining \(n - 1\) columns.
Is GE Good Enough?

A sparse matrix is “any matrix with enough zeros that it pays to take advantage of them.” — J. Wilkinson

- For dense problems (almost all entries non-zero), yes.
- But what about sparse problems?
- Example: 1D Heat equation has 3 non-zeros per row.

1D Heat Equation Sparsity
Introducing Iterative Methods

\[ Ax = b \]

- Idea: Sparse matrix-vector products are cheap
cost = \# non-zeros.
- Let \( D = \text{diag}(A) \) contain “a lot” of the matrix. Then,

\[
(D + (A - D))x = b
\]

\[
Dx = b - (A - D)x
\]

\[
x = D^{-1}(b - (A - D)x)
\]

- Jacobi’s method:

\[
x_{i+1} = x_i + D^{-1}(b - Ax_i)
\]

- Total Operations \( \approx \text{nnz.} \)
Consider a model Laplace problem of size: \( n = k^d \), where \( d = 2, 3 \).

<table>
<thead>
<tr>
<th>Method</th>
<th>2D</th>
<th>3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dense GE</td>
<td>( k^6 )</td>
<td>( k^9 )</td>
</tr>
<tr>
<td>Sparse GE</td>
<td>( k^3 )</td>
<td>( k^6 )</td>
</tr>
<tr>
<td>Jacobi</td>
<td>( k^4 \log k )</td>
<td>( k^5 \log k )</td>
</tr>
</tbody>
</table>

Table from: *Scientific Computing: An Introductory Survey*, 2nd ed. by M.T. Heath
Speed of Various Methods

Consider a model Laplace problem of size: \( n = k^d \), where \( d = 2, 3 \).

<table>
<thead>
<tr>
<th>Method</th>
<th>2D</th>
<th>3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dense GE</td>
<td>( k^6 )</td>
<td>( k^9 )</td>
</tr>
<tr>
<td>Sparse GE</td>
<td>( k^3 )</td>
<td>( k^6 )</td>
</tr>
<tr>
<td>Jacobi</td>
<td>( k^4 \log k )</td>
<td>( k^5 \log k )</td>
</tr>
<tr>
<td>Multigrid</td>
<td>( k^2 )</td>
<td>( k^3 )</td>
</tr>
</tbody>
</table>

Table from:

*Scientific Computing: An Introductory Survey*, 2nd ed. by M.T. Heath
Outline

- Background.
- Computation in the Physical Sciences.
- Solving Linear Systems with Iterative Methods.
  - Introduction Multilevel Methods.
- Open Questions in Multilevel Methods.
Introducing Multilevel Methods

- Goal: Solve problem with specified mesh spacing, $h$.
- Idea: Approximate problem w/ coarse mesh $H$.

- Big Question: Will this work?
Fourier Series

Consider a (real) Fourier series

\[ f(x) = \frac{a_0}{2} + \sum_{i=1}^{\infty} \alpha_i \cos(2\pi xi) \]

What do these functions look like?

Smooth

Oscillatory
Sampling Fourier Modes

What modes can a discretization sample?
Question: What does this have to do with multigrid?

- Coarse grids can only resolve smooth modes.
- Coarse grids cannot resolve oscillatory modes (aliasing).

Next question: What about oscillatory modes?

Coarse Grid OK. Coarse Grid no help.
Jacobi to the Rescue

Introduction to Multilevel Solvers for the Physical Sciences – p.32/45
Multigrid by Picture

Smooth Smooth Smooth Smooth
Solve
Multigrid Method for $A_h x = b$

Loop until convergence...

1. Smooth on fine grid.
   \[ \text{jacobi}(A_h, x, b). \]

2. Transfer residual $(b - A_h x)$ to coarse grid (restriction).
   \[ r_c = P^T(b - A_h x). \]

3. Solve on coarse grid.
   \[ x_c = A_H^{-1} r_c. \]

4. Transfer solution to fine grid (prolongation).
   \[ x = x + P x_c \]

5. Smooth on fine grid.
   \[ \text{jacobi}(A, x, b). \]
Outline

- Background.
- Computation in the Physical Sciences.
- Solving Linear Systems with Iterative Methods.
- Introduction to Multilevel Methods.
- Open Questions in Multilevel Methods.
Open Questions in Multigrid

- MG is designed problems like Laplace or Heat equation.
- On other problems additional issues arise.
- Mathematical issues: anisotropy, systems, variable materials.
- Computer science issues: parallelism, scalability.
Math Issue #1: Anisotropy

\[
\frac{\partial^2 u}{\partial x^2} + \epsilon \frac{\partial^2 u}{\partial y^2} = f
\]

- Anisotropic operators have direction-dependent behavior.
- Example: Heat diffuses “faster” in \( y \) direction (\( \epsilon \) small).
- Tests varying \( \epsilon \) w/ 10,000 unknowns.

<table>
<thead>
<tr>
<th>( \epsilon = 1 )</th>
<th>( \epsilon = 10^{-1} )</th>
<th>( \epsilon = 10^{-2} )</th>
<th>( \epsilon = 10^{-3} )</th>
<th>( \epsilon = 10^{-4} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations</td>
<td>14</td>
<td>20</td>
<td>53</td>
<td>129</td>
</tr>
</tbody>
</table>

- This is BAD!
Reacting to Anisotropy

- Better meshes fix some problems.

![Isotropic Mesh](image1) ![Anisotropic Mesh](image2)

- Meshes alone cannot solve hard problems.
- Research problem: Robust detection of anisotropy.
- Research problem: Non-axial anisotropy.
Math Issue #2: PDE Systems

- PDE systems multiple different types of variables (e.g. displacement, velocity, pressure, temperature, etc.).
- Example: Linear elasticity.
- One solution: Smoothed aggregation — explicitly preserve null space on coarse levels.
- Research problem: Fluid problems (e.g. Navier-Stokes).

Image courtesy of the CSAR/UIUC
http://www.csar.uiuc.edu
Math Issue #3: Multimaterial

- Material interfaces can be sites of discontinuities ⇒ oscillatory modes at boundaries.
- Features can be hard to resolve on coarse grid.

- Research problem: Detecting material interfaces.
- Research problem: Handling disappearing features.
CS Issues: Parallelism

- More processors *should* lead to faster solutions.
- Strong scaling — fix work, increase processors.
- Example: 2,000 steps of Jacobi.
CS Issues: Parallelism

- More processors *should* lead to faster solutions.
- Strong scaling — fix work, increase processors.
- Example: 2,000 steps of Jacobi.

Question: What causes the loss in efficiency?
Understanding Efficiency

- Answer: Computation to communication ratio.
- Weak scaling — fix work per processor.

Message: What works on a small # of procs, might not work on a large #.
CS Issue #1: Scalability

- Coarse grids ⇒ less work per proc ⇒ poor performance.
- One solution: Move data to leave some procs idle.
- Research problem: What is the best way to repartition?
- Research problem: How to address poor performance on really big (terascale) computers.

Red Storm (SNL) 26,569 procs
Jaguar (ORNL) 23,016 procs
Take Home

“I would rather have today’s algorithms on yesterday’s computers than vice versa.” - Reported by P. Toint

- Ubiquity of computational science.
- Importance of good algorithms.
- Rationale behind multilevel algorithms.
- Nature of the “big questions” in multilevel algorithm research.
  - Math: Anisotropy, multimaterial, PDE systems.
  - CS: parallelism, scalability.