SAND2007-7657P

Introduction to Multilevel Solvers for the Physical Sciences

Chris Siefert

Scalable Algorithms Group

Sandia National Laboratories

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

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.
  - Trilinos project: `http://trilinos.sandia.gov`

Sandia National Laboratories

Course Background

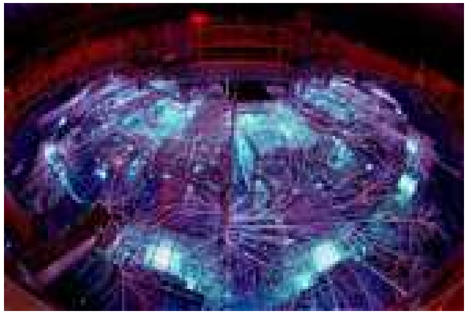
- Assumed audience background:
  - Multivariable calculus (MATH 212).
  - Linear algebra (MATH 211).
- A more detailed talk would require:
  - Algorithms (CS 303).
  - Advanced linear algebra (MATH 408).
  - Numerical analysis (MATH 413, 414).

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?

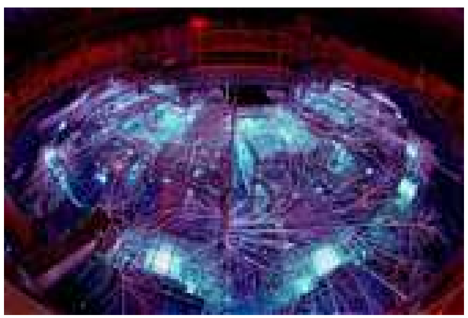
- What do we think of when we think of computational science?
  - Usually “big” things. . .
  - Airplanes, cars, rockets, etc.



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.
- 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:

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

Sandia National Laboratories

Example: Finite Differences (2)

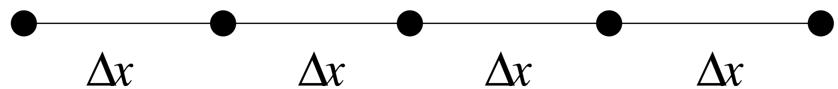
- Model:

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

- Discretization — Backward Euler (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:



Solve the Discrete Problem

"Mathematics is the queen of the sciences" – Carl Friedrich Gauss

- Backward Euler (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:

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

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 change 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
  $\Rightarrow$ One linear solve per time step!
- We can do this w/ Gaussian elimination.
- 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.

Sandia National Laboratories

Gaussian Elimination

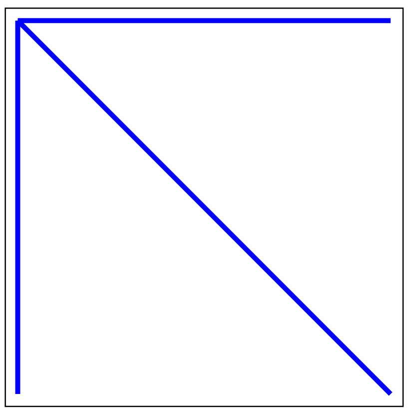
$$\begin{bmatrix} 1 & b & c \\ 0 & 1 & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}$$

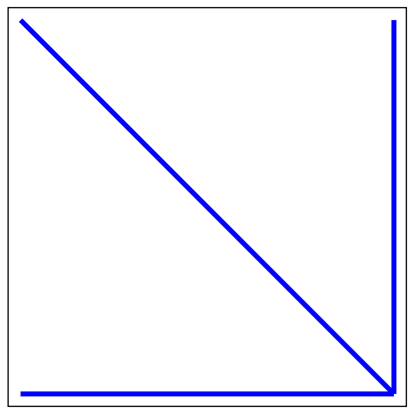
Total Operations $\approx n^3$

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 Gauss Good Enough?

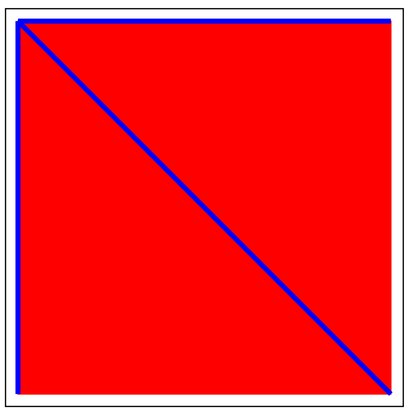
- For dense problems (almost all entries non-zero), yes.
- But what about sparse problems?
- Example: 1D Heat equation has 3 non-zeros per row.
- Sparse GE is better, but not good enough
  $\Rightarrow$ work is heavily dependent on matrix structure.



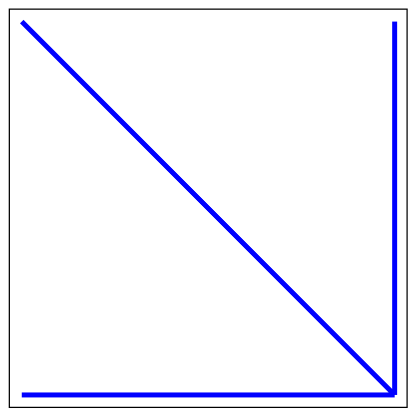


Is Gauss Good Enough?

- For dense problems (almost all entries non-zero), yes.
- But what about sparse problems?
- Example: 1D Heat equation has 3 non-zeros per row.
- Sparse GE is better, but not good enough
  $\Rightarrow$ work is heavily dependent on matrix structure.



n^3 work $\longrightarrow$ BAD!



n work $\longrightarrow$ GOOD!

Introducing Iterative Methods

$$Ax = b$$

- Idea: Sparse matrix-vector products are cheap
  cost = # non-zeros.
- This is the basic idea behind iterative methods.
- Jacobi's method:

  $$x_{i+1} = D^{-1}(b - (A - D)x_i)$$

  where D is the diagonal of A.
- Total Operations $\approx bn$ per step, where b = avg. nnz per row.

Sandia National Laboratories

Speed of Various Methods

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

Method	2D	3D
Dense GE	k^6	k^9
Sparse GE	k^3	k^6
Jacobi	$k^4 \log k$	$k^5 \log k$

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 $d = 2, 3$.

Method	2D	3D
Dense GE	k^6	k^9
Sparse GE	k^3	k^6
Jacobi	$k^4 \log k$	$k^5 \log k$
Multigrid	k^2	k^3

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 with solution of problem with coarse mesh H, where $H > h$.
- How to move from coarse (H) to fine (h) $\Rightarrow$ interpolation.
- How to solve coarse (H) problem $\Rightarrow$ GE, or more multigrid.
- Use Jacobi to clean up the rest.

- Big Question: Will this work?

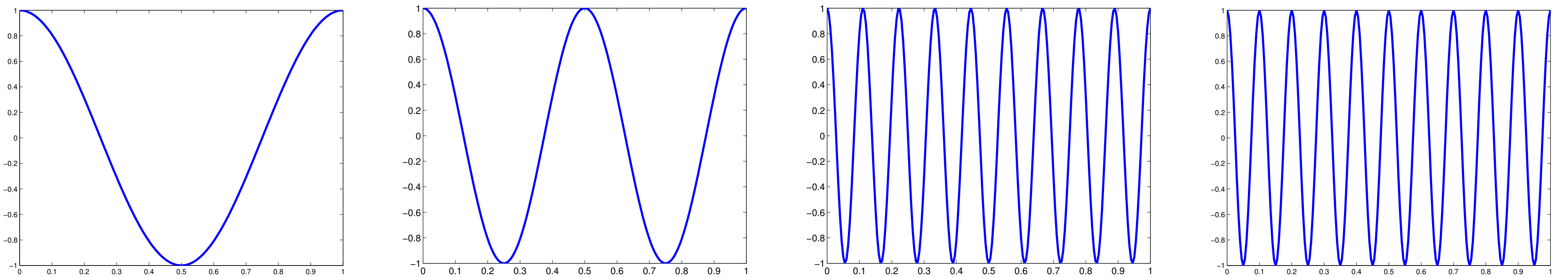
Sandia National Laboratories

Fourier Series

- Consider a (real) Fourier series

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

- What do these functions look like?

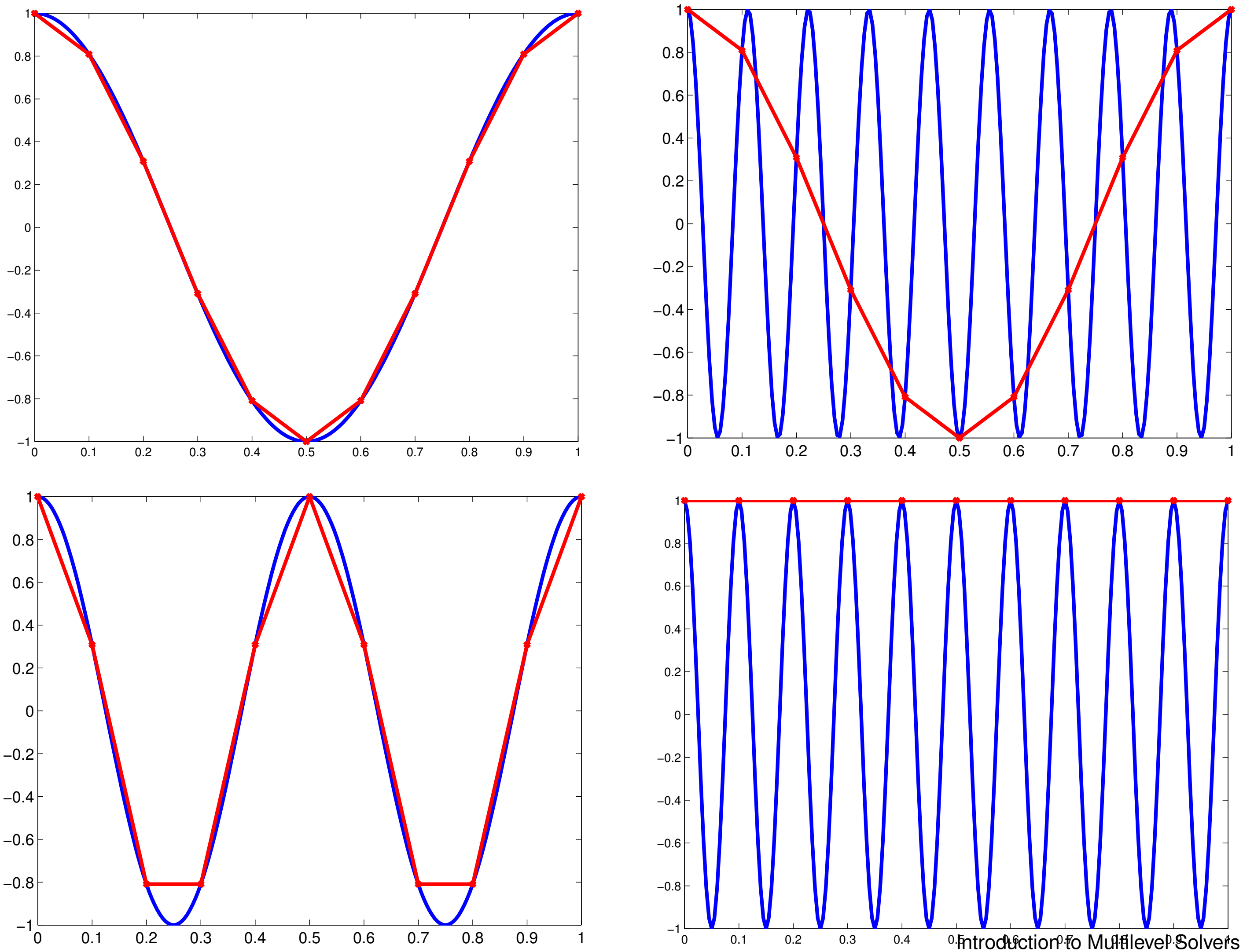


Smooth Oscillatory

Sandia National Laboratories

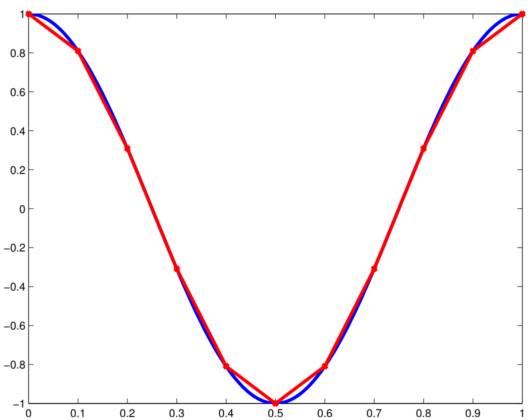
Sampling Fourier Modes

- What modes can a discretization sample?

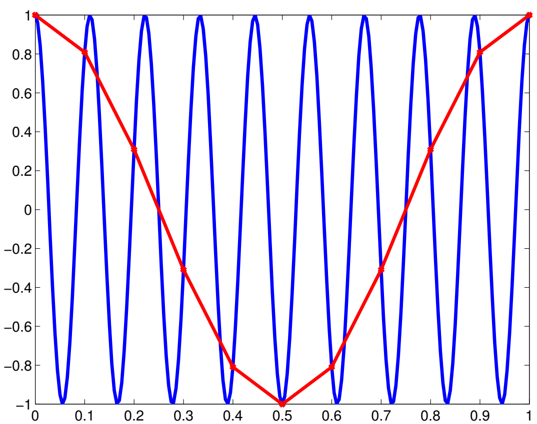


Multigrid & Fourier Modes

- 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?

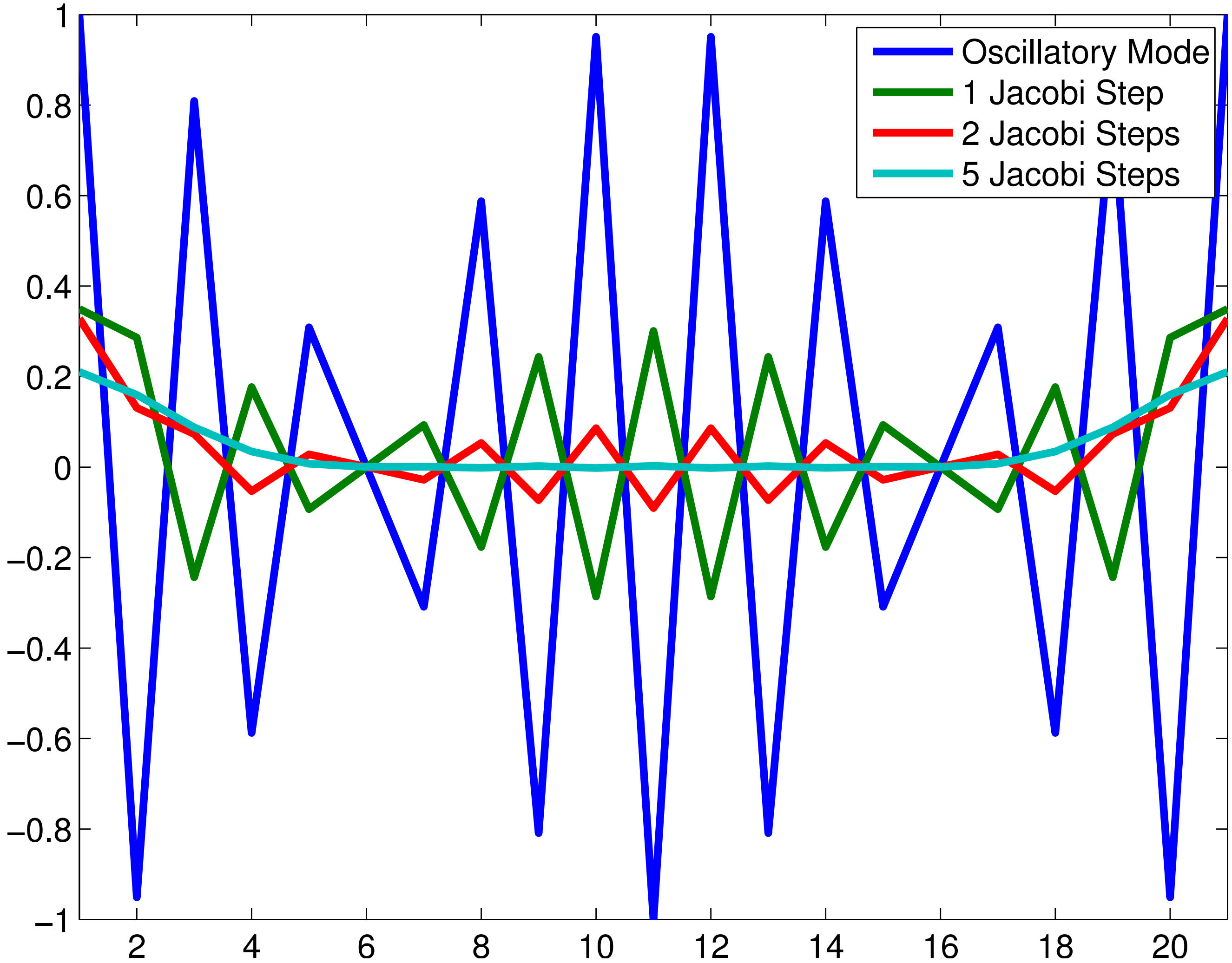


Coarse Grid OK.

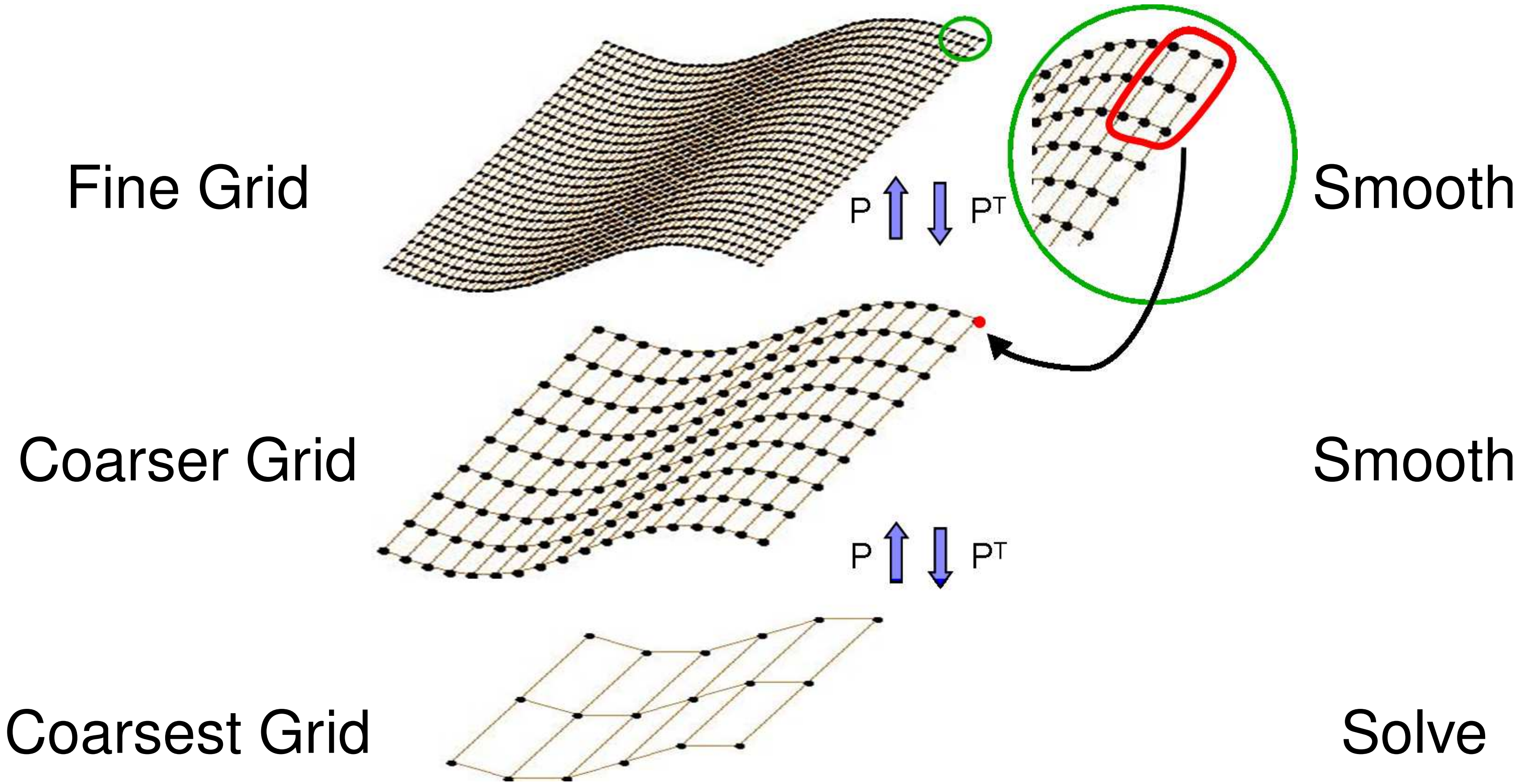


Coarse Grid no help.

Jacobi to the Rescue



Multigrid by Picture



Multigrid Method for $A_h x = b$

Loop until convergence...

1. Smooth on fine grid.
   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.
   jacobi$(A_, x, b)$.

Sandia National Laboratories

Algebraic Multigrid

- Previous method is known as Geometric Multigrid
  $\Rightarrow$ New discretization required for each H.
- This is not necessary...we can do this algebraically.
- Algebraic Multigrid (AMG)
  - Only needs matrix A_h.
  - Generates grid transfer (P) by grouping variables together.
  - Coarse matrix formed by matrix-matrix multiplication $A_H = P^T A_h P$.

Outline

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

Sandia National Laboratories

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.

Sandia National Laboratories

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 (ϵ small).
- Tests varying ϵ w/ $10,000$ unknowns.

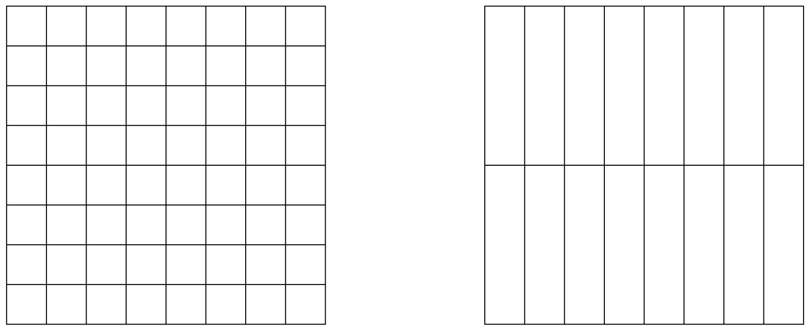
	$\epsilon = 1$	$\epsilon = 10^{-1}$	$\epsilon = 10^{-2}$	$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$
Iterations	14	20	53	129	189

- This is BAD!

Sandia National Laboratories

Reacting to Anisotropy

- Better meshes fix simple problems

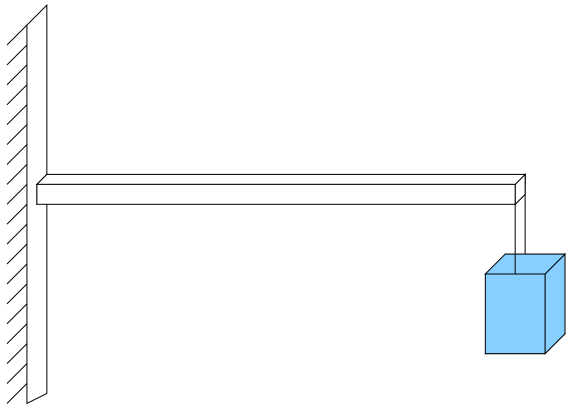


Isotropic Mesh   Anisotropic Mesh

- Meshes alone cannot solve hard problems.
- One solution: Hot-dog shaped aggregates (change coarse operator).
- Research problem: Robust detection of anisotropy.
- Research problem: Non-axial anisotropy.

Math Issue #2: PDE Systems

- PDE systems have more variables and larger null spaces.
- Example: Linear elasticity.

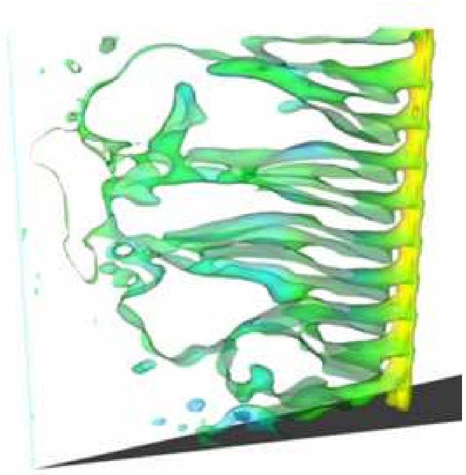


- One solution: Smoothed aggregation — explicitly preserve null space on coarse levels.
- Research problem: Fluid problems (e.g. Navier-Stokes).

Sandia National Laboratories

Math Issue #3: Multimaterial

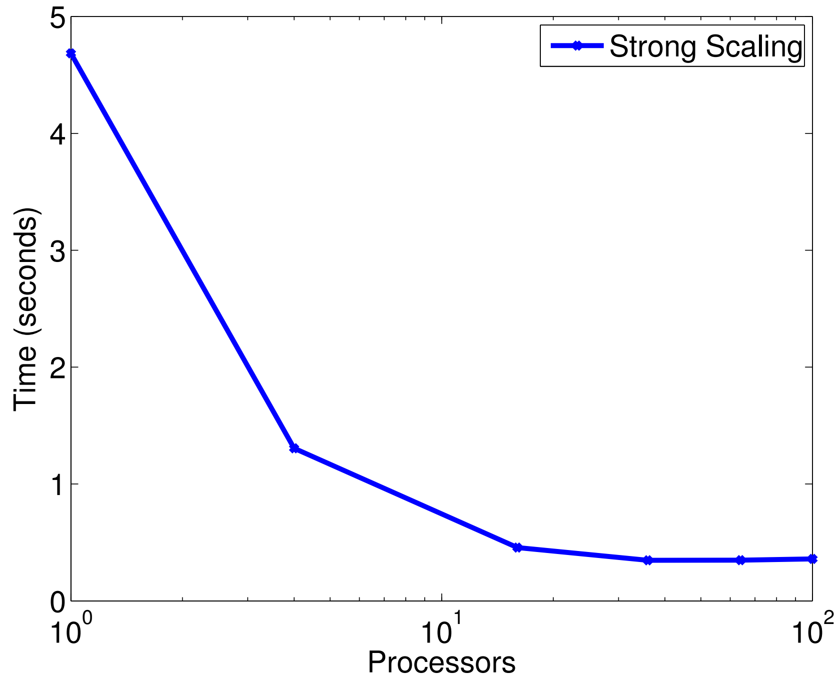
- Material interfaces can be sites of discontinuities $\Rightarrow$ oscillatory modes at boundaries.
- Features can be hard to resolve on coarse grid.



- Research problem: Detecting material interfaces.
- Research problem: Handling disappearing features.

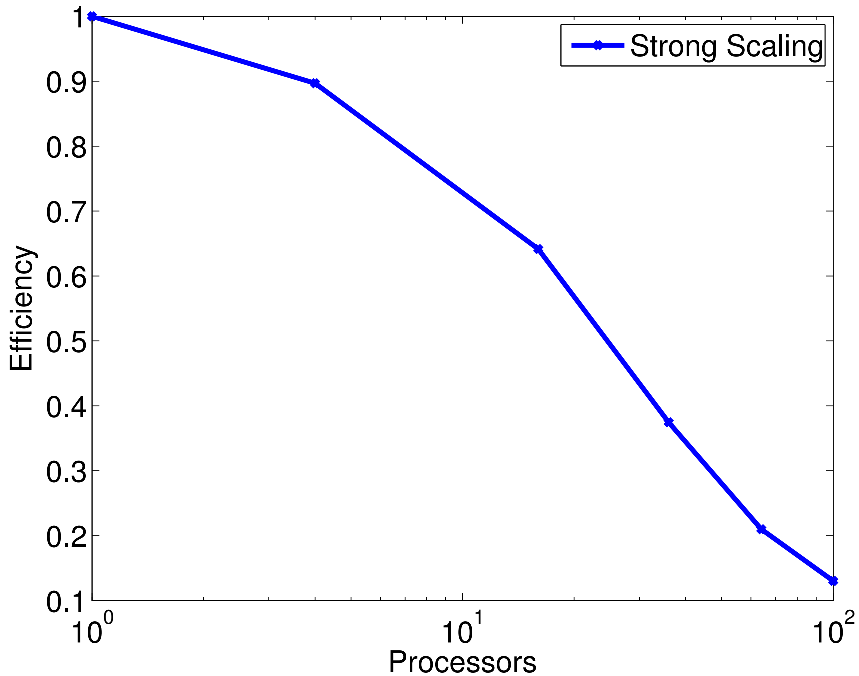
CS Issues: Parallelism

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



CS Issues: Parallelism

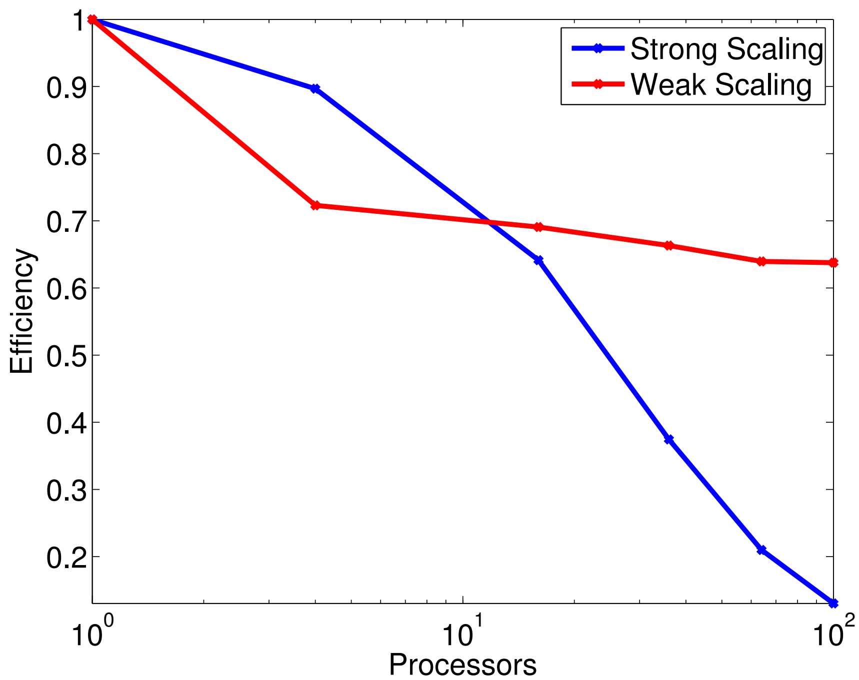
- More processors *should* lead to faster solutions.
- Strong scaling — fix work, increase processors.
- Example: 2,000 steps of Jacobi.



- 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: Repartitioning to purposely 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.

Sandia National Laboratories

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.

Sandia National Laboratories

Useful References

- *A Multigrid Tutorial*, 2nd ed. W.L. Briggs, V.E. Henson and S.F. McCormick.
- *Scientific Computing: An Introductory Survey*, 2nd ed. M.T. Heath.
- *Numerical Solution of Partial Differential Equations.* K.W. Morton and D.R. Mayers.

- Trilinos project: `http://trilinos.sandia.gov`
- My web site: `http://www.sandia.gov/~csiefer`