Numerical Methods

CS 357 Fall 2013

David Semeraro

Science Application

- Partial differential equations.
- Solution by finite difference methods.
- Numerics
  - Derivative approximation (Taylor Series)
  - Linear Algebra (Sparse Matrices)
  - Iterative methods (Jacobi)
  - Convergence (Eigenvalues)
  - Numerical error (effect of denormal numbers)

Laplace's Equation

- Elliptic PDE.
  - Heat transfer
  - Incompressible viscous flow
- $\nabla^2\varphi = \frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} = 0$ in domain Ω
- Boundary Conditions $\varphi(x, y) = f(x, y)$ on δ(Ω)

Laplace's Equation

- Solution Domain

$$\Omega = \begin{cases} 0 < x < 1 \\ 0 < y < 1 \end{cases}$$

- Boundary Conditions

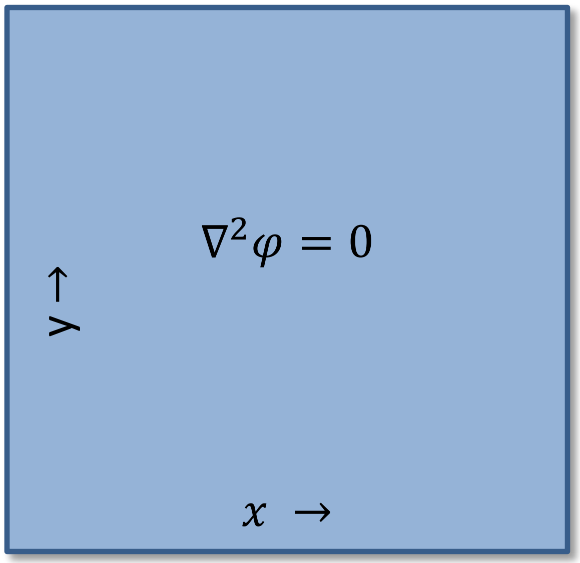
$$\varphi(x,y) = 0 \begin{cases} x = 0; 0 \leq y \leq 1 \\ x = 1; 0 \leq y \leq 1 \\ y = 1; 0 \leq x \leq 1 \end{cases}$$

$$\varphi(x,y) = \sin(2\pi x)\{y = 0; 0 \leq x \leq 1$$

Laplace’s Equation

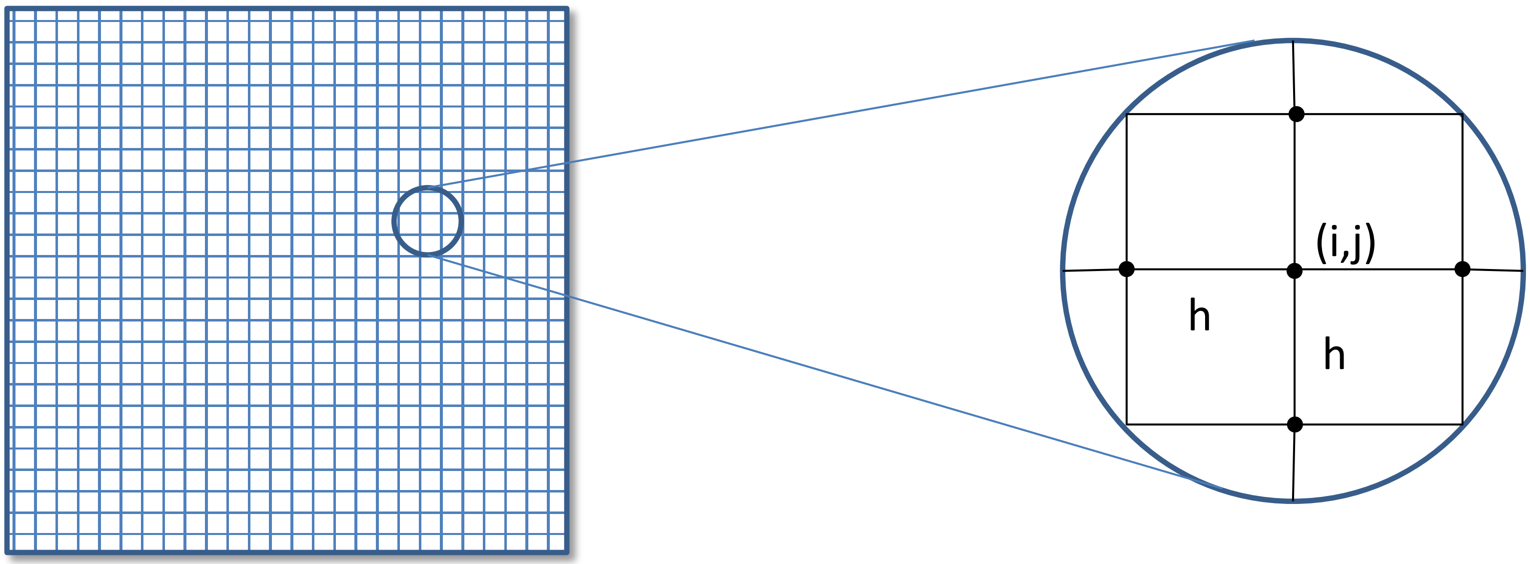
$$\varphi = 0$$

$\varphi = 0$ $\qquad$ $\varphi = 0$



$$\varphi = \sin(2\pi x)$$

Finite Difference Grid



- Approximate the function φ at specific points in the domain Ω.
- $x = i \times h, y = j \times h$
- $0 < i < n; 0 < j < n$ in domain Ω
- $h = \frac{1}{n}$
- Use Taylor series to approximate PDE at (i,j).

Approximate Derivatives

- Taylor series approximation of function on computational mesh.
  - X direction.
    - $\varphi(x+h,y) = \varphi(x,y) + h\varphi'(x,y) + \frac{h^2}{2}\varphi''(x,y) + \frac{h^3}{3!}\varphi'''(x,y) + O(h^4)$
    - $\varphi(x-h,y) = \varphi(x,y) - h\varphi'(x,y) + \frac{h^2}{2}\varphi''(x,y) - \frac{h^3}{3!}\varphi'''(x,y) + O(h^4)$

Approximate Derivatives

- $\varphi(x+h,y)+\varphi(x-h,y)=2\varphi(x,y)+h^2\varphi''(x,y)+O(h^4)$
- $\frac{\partial^2\varphi}{\partial x^2}=\frac{1}{h^2}[\varphi(x+h,y)-2\varphi(x,y)+\varphi(x-h,y)]+O(h^2)$
- $\frac{\partial^2\varphi}{\partial x^2}\approx\frac{1}{h^2}[\varphi(x+h,y)-2\varphi(x,y)+\varphi(x-h,y)]$

Perform similar operations to approximate $\frac{\partial^2\varphi}{\partial y^2}$.

Approximate Derivatives

- We now have approximations to the derivatives at (x, y).
- We introduce an new variable valid at the node points that approximates the value of φ at the node points.

$$\hat{\varphi}(i, j) \approx \varphi(x, y)$$
$$x = i \times h$$
$$y = j \times h$$

Finite Difference Approximation

- Combining derivative approximations gives:

$$\nabla^2\varphi(x,y) \approx \frac{1}{h^2}[\hat{\varphi}(i-1,j)+\hat{\varphi}(i,j-1) + \hat{\varphi}(i+1,j)+\hat{\varphi}(i,j+1)-4\hat{\varphi}(i,j)] = 0$$

- Satisfied at *interior* points of the domain.
- The error in the approximation is $O(h^2)$.

Linear Algebra

- Application of the approximation to the Laplace equation at each of the interior points in the grid yields a large sparse linear system of equations in the unknowns $\hat{\varphi}(i, j)$.
- The system is block tridiagonal.

Linear Algebra

- The linear system of equations has the following bock form.

$$\frac{1}{h^2}\begin{bmatrix} D & I & & \\ I & D & \ddots & \\ & \ddots & \ddots & I \\ & & I & D \end{bmatrix}\begin{bmatrix} \hat{\varphi}(i,1) \\ \hat{\varphi}(i,2) \\ \vdots \\ \hat{\varphi}(i,n-1) \end{bmatrix} = \frac{1}{h^2}\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \end{bmatrix}$$

- The elements of b_i are functions of the boundary conditions.
- The blocks $D \ \in \ R^{n-1\times n-1}$

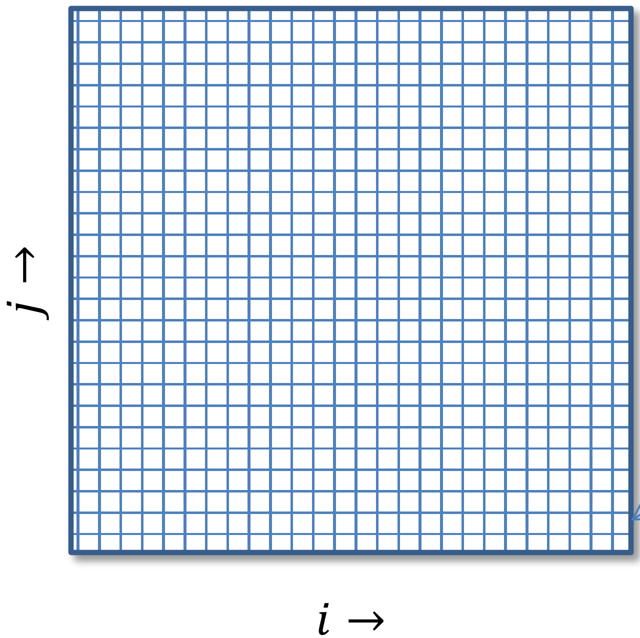
Linear Algebra

- The block D is tridiagonal with the following structure.

$$\begin{bmatrix} -4 & 1 & & \\ 1 & -4 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -4 \end{bmatrix}$$

- The blocks in the partition correspond to rows in the computational mesh.

Linear Algebra



Elements of $\hat{\varphi}$ are ordered with i increasing fastest. (arbitrary decision)

$\hat{\varphi}(i, 1)$ corresponds to first row in the mesh

Iterative Methods

- We have transformed our PDE into a linear system of equations. (Continuous to Discrete).
- Direct solution is expensive and inefficient compared to iterative methods.
- Explore some alternatives.
  - Jacobi
  - GS
- The eigenvalues of the jacobi and GS iteration matrices are all < 1. However the coefficient matrix is not strictly diagonally dominant.

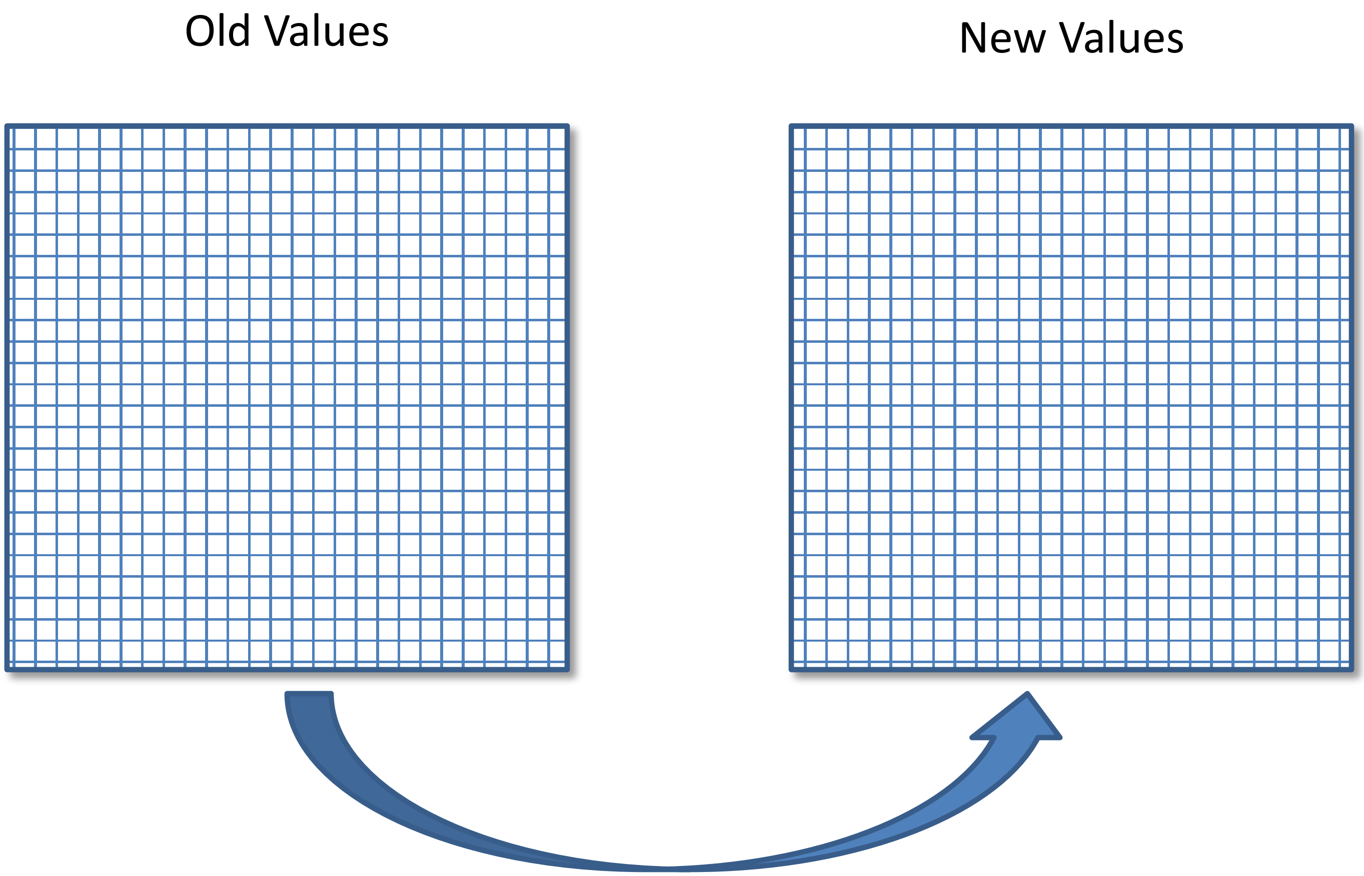
Jacobi's Method

```
For (j=1;j<n;j++):
        for(i=1;i<n;i++):
```

$$\varphi_{i,j}^{k+1} = (\varphi_{i-1,j}^{k} + \varphi_{i+1,j}^{k} + \varphi_{i,j-1}^{k} + \varphi_{i,j+1}^{k})/4$$

- Move along the mesh from bottom to top and from left to right forming new values of φ.
- Do not use the new values until the entire mesh has been updated.

Jacobi's Method



GS Method

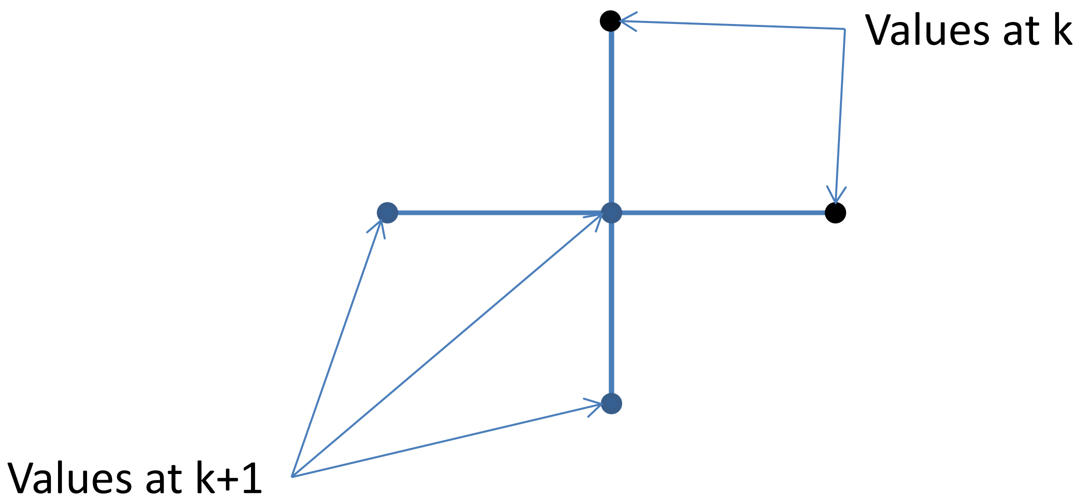
For (j=1;j<n;j++):

for(i=1;i<n;i++):

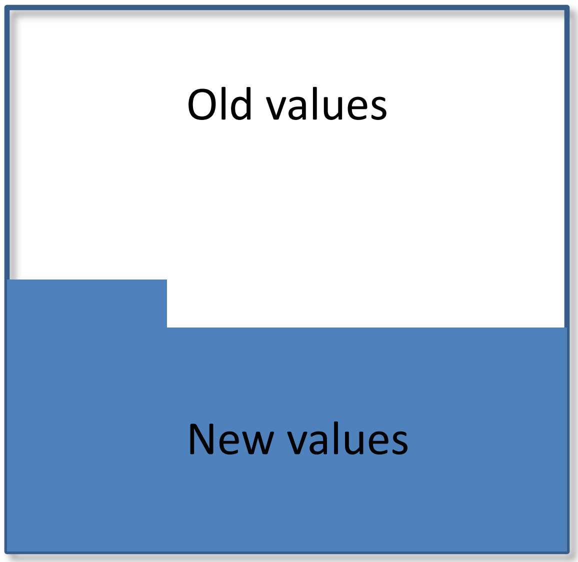
$$\varphi_{i,j}^{k+1} = (\varphi_{i-1,j}^{k+1} + \varphi_{i+1,j}^{k} + \varphi_{i,j-1}^{k+1} + \varphi_{i,j+1}^{k})/4$$

- Move along the mesh from bottom to top and from left to right forming new values of φ.
- Use the new values of φ as soon as they are available.

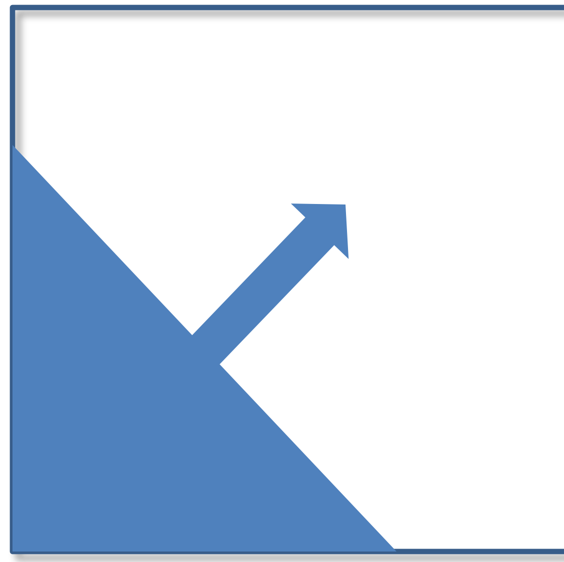
GS Method



GS Method



Classic GS goes left to right, bottom to top.



There is enough information along the wave front to evaluate the new values in parallel

Block Methods

- Take advantage of the block nature of the coefficient matrix to advance the solution faster.
- Use permutations of the rows and columns and the nature of the stencil to form new methods. (red black reordering)

Block Jacobi

$$\begin{bmatrix} D & I & & \\ I & D & \ddots & \\ & \ddots & \ddots & I \\ & & I & D \end{bmatrix} \begin{bmatrix} \hat{\varphi}(i,1) \\ \hat{\varphi}(i,2) \\ \vdots \\ \hat{\varphi}(i,n-1) \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \end{bmatrix}$$

$$\hat{\varphi}_{i,j}^{k+1} = D^{-1}\left[b_j - \hat{\varphi}_{i,j-1}^{k} - \hat{\varphi}_{i,j+1}^{k}\right]$$

Block GS

$$\begin{bmatrix} D & I & & \\ I & D & \ddots & \\ & \ddots & \ddots & I \\ & & I & D \end{bmatrix} \begin{bmatrix} \hat{\varphi}(i,1) \\ \hat{\varphi}(i,2) \\ \vdots \\ \hat{\varphi}(i,n-1) \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \end{bmatrix}$$

$$\hat{\varphi}_{i,j}^{k+1} = D^{-1}\left[b_j - \hat{\varphi}_{i,j-1}^{k+1} - \hat{\varphi}_{i,j+1}^{k}\right]$$

Block Methods

- Solve a tridiagonal matrix problem at each step.
  - the matrix D can be factored once and used for each pass
- Jacobi advances the solution one row at a time not using the new values in theprevious row.
- GS advances the solution but uses the new values in the previous row.

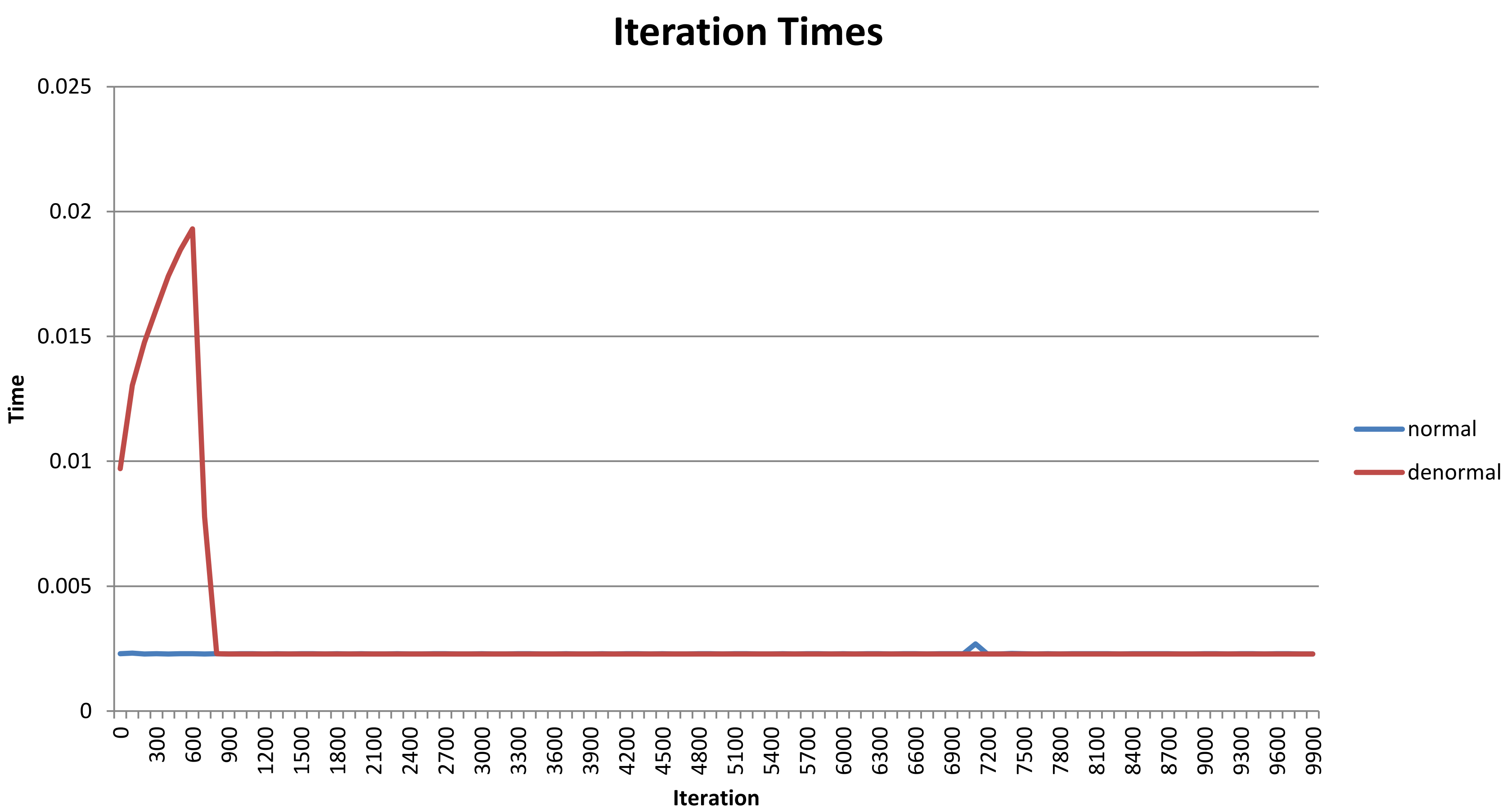
Alternating Direction Methods

- Apply GS from bottom to top and then reverse direction and go from top to bottom.
- Sweep GS from left to right and then from right to left operating on the columns of the computational mesh rather than the rows.
- Methods are called Alternating Direction Implicit (ADI) methods.

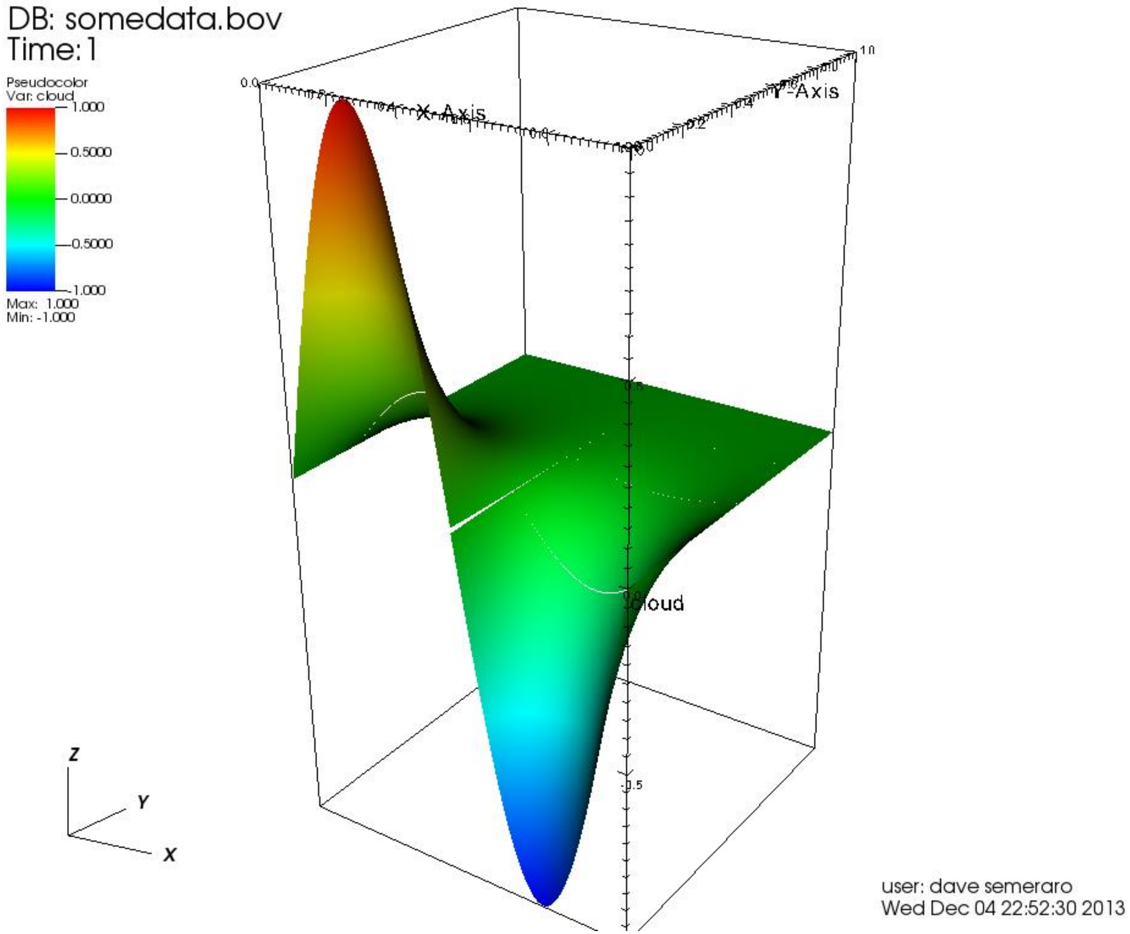
Effect of Denormal Numbers

- Recall denormal number are floating point numbers smaller than machine epsilon.
- Care must be taken in how computational grid is initialized.
  - Initialize to zero
  - Initialize to some number larger than mach eps.
- Early computations in corners of zero initialized grid cause denormal numbers to appear.

Effect of Denormal Numbers



Solution


DB: somedata.bov
Time:1
Pseudocolor
Var: cloud
1.000
0.5000
0.0000
-0.5000
-1.000
Max: 1.000
Min: -1.000
X-Axis
Y-Axis
cloud
Z
Y
X
user: dave semeraro
Wed Dec 04 22:52:30 2013