An Iterative Solver for the Diffusion Equation

Alan Davidson

adavidso@cs.hmc.edu

25 April 2006

The Diffusion/Heat Equation

$$u_t = a + D \cdot u_{xx}$$

- u is the concentration/temperature
- a is a source/sink
- D is a diffusion/thermal diffusivity constant
- t is time, x is space

The Hitch

- For clinic, we needed arbitrary Dirichlet boundary conditions through the middle
- These BCs simply hold the concentration at a fixed amount
- Exact solution cannot be found easily

The Solution!

- Use an iterative solver
- For clinic we actually used
  - The 3-D equation
  - The Gauss-Seidel method
  - The backwards Euler FDA
  - C++
- I wanted to try
  - 1, 2, or 3 dimensions
  - Dirichlet, Neumann, and Cauchy BCs
  - The Jacobi or SSOR methods
  - The backwards Euler FDA
  - Sparse matrices in Matlab

Iterative Methods

We want to solve

$$A\vec{x} = \vec{b}$$

but we don't want to invert A (time constraints, etc). Divide up A so that

$$A = D + L + U$$

where

- D is diagonal
- L is lower triangular
- U is upper triangular

The Jacobi Method

Pick $\vec{x}^{(0)}$ to be the initial guess at a solution. Now, define

$$\vec{x}^{(i+1)} = D^{-1} \cdot (-L - U) \cdot \vec{x}^{(i)} + D^{-1} \cdot \vec{b}$$

If $\left|\vec{x}^{(i+1)} - \vec{x}^{(i)}\right|$ isn't small enough, repeat.

The Successive Over-Relaxation (SOR) Method

We will solve

$$\omega A\vec{x} \;=\; \omega\vec{b}$$

Noting that

$$\omega A \;=\; (D + \omega L) + (\omega U - (1-\omega)D),$$

we now have that

$$\vec{x}^{(i+1)} \;=\; (D+\omega L)^{-1} \cdot \left((-\omega U + (1-\omega)D)\vec{x}^{(i)} + \omega\vec{b}\right)$$

The backwards version is

$$\vec{x}^{(i+1)} \;=\; (D+\omega U)^{-1} \cdot \left((-\omega L + (1-\omega)D)\vec{x}^{(i)} + \omega\vec{b}\right)$$

The Symmetric Successive Over-Relaxation (SSOR) Method

We do one forwards SOR step followed by one backwards SOR step:

$$\begin{aligned} \vec{x}^{(i+1/2)} &= (D + \omega L)^{-1} \cdot \left((-\omega U + (1-\omega) D) \vec{x}^{(i)} + \omega \vec{b} \right) \\ \vec{x}^{(i+1)} &= (D + \omega U)^{-1} \cdot \left((-\omega L + (1-\omega) D) \vec{x}^{(i+1/2)} + \omega \vec{b} \right) \end{aligned}$$

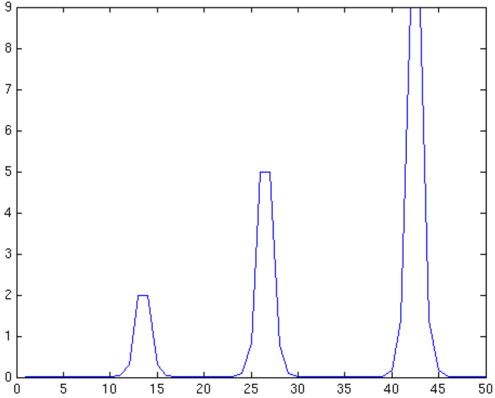
Progress So Far...

Slower than I expected

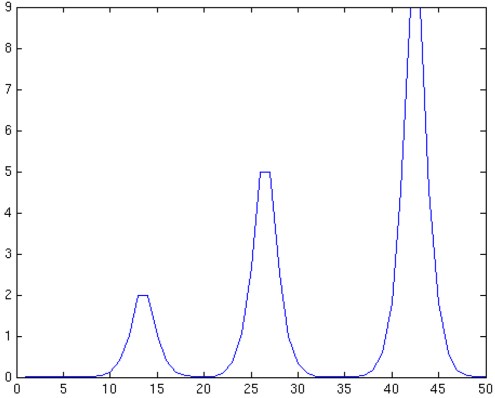
- 1-D case implemented for both Jacobi and SSOR methods with any Dirichlet BCs
- Neumann (and therefore Cauchy) BCs are not well defined in arbitrary locations, especially in a 1-D case
- Uses only sparse matrices
- Checks for stability

Oddly enough, the Jacobi method seems to converge more quickly than the SSOR!?

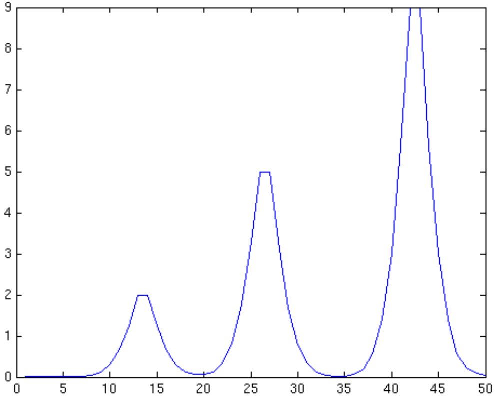
An Example Run...



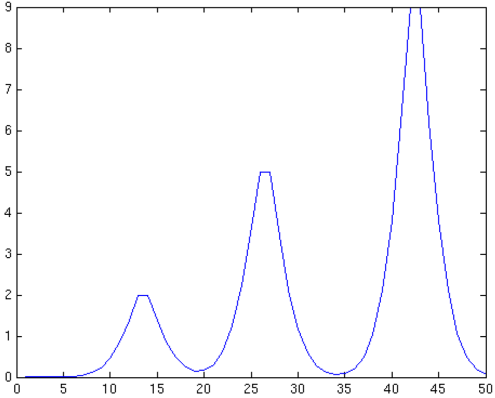
An Example Run...



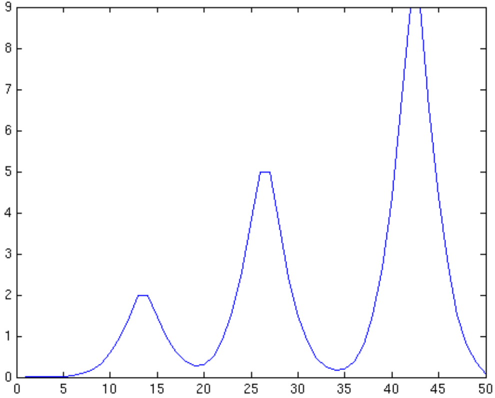
An Example Run...



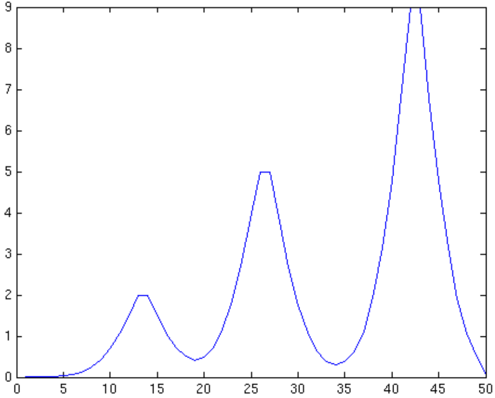
An Example Run...



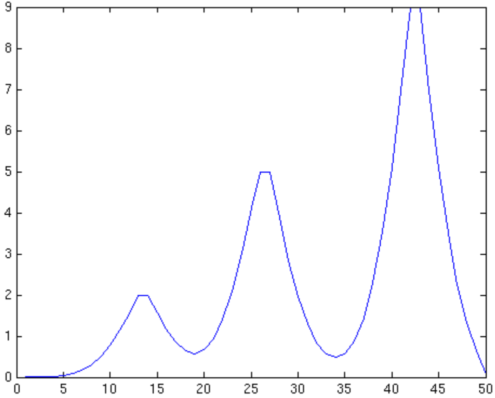
An Example Run...



An Example Run...



An Example Run...



References Used

Saad, Yousef. Iterative Methods for Sparse Linear Systems SIAM, 2000.

http://mathworld.wolfram.com

That's All, Folks!

Questions?