Nick Travers

228B Lecture Notes 01/29/09

Recall we are trying to solve the the heat equation $u_t = bu_{xx}$. Using the Crank-Nicolson discretization we get

$$(I - \frac{b\Delta t}{2}L)u^{n+1} = (I + \frac{b\Delta t}{2}L)u^n \tag{1}$$

In 1D this can be solved easily and directly using gaussian elimination since we have a tri-diagonal matrix. However, in higher dimensions this is NOT the case. Last time we discussed possibly using CG, SOR, or MG. All of these should work, but what sort of convergence can we expect?

We Consider Two Extreme Cases:
(1) $\Delta t \to 0$, h fixed
(2) $\Delta t \to \infty$, h fixed

case (1):
If $\Delta t \to 0$ then $I - \frac{b\Delta t}{2}L \to I$
I is, of course, very easy to invert so we expect convergence in 1 step from an iterative method.

case (2):
If $\Delta t \to \infty$ then $I - \frac{b\Delta t}{2}L$ blows up. But we are actually trying to solve:

$$(I - \frac{b\Delta t}{2}L)u^{n+1} = (I + \frac{b\Delta t}{2}L)u^n$$
$$-\frac{b\Delta t}{2}Lu^{n+1} = \frac{b\Delta t}{2}Lu^n + o(1/\Delta t)$$
$$-Lu^{n+1} = Lu^n = r$$

This is just the poisson equation. So at best our iterative methods (CG,SOR,MG) take 1 iteration, and at worst they take the same number of iterations as required to solve the poisson equation.

Why is using $\Delta t \to \infty$ interesting? When using Crank-Nicolson and taking $\Delta t \to 0$ with $\Delta t/h$ constant then $\|L\| \to \infty$ ($L \sim 1/h^2$). Thus, $\|\Delta t L\| \to \infty$ as $h \to 0$ ($\Delta t L \sim 1/h$). So again we are essentially solving the poisson equation at each time step.

As $\Delta t \to 0$, with $\Delta t/h$ fixed, the condition number of the matrix $A = I - \frac{b\Delta t}{2}L$ increases from that of the identity I (small) to that of the discrete laplacian L (large). Does this suggest that convergence will slow down? Not necessarily since we have a good initial guess $u^{n+1} \sim u^n$ and this approximation gets even better as $\Delta t \to 0$.

Main Point
All iterative methods (CG,SOR,MG) for solving Laplace's equation generally work even better when solving the diffusion equation because:

1. The matrix is better conditioned
2. We have a much better initial guess $u^{n+1} \sim u^n$.

Example
Using MG to solve $\Delta u = f(x)$ or $b\Delta u = u_t$ on a 64 by 64 grid with periodic BC, $\nu_1 = \nu_2 = 1$ and GS-RB as a smoother.

(1) <u>Possion Equation</u>: convergence factor $\rho = 0.16$, 7-8 iterations to reduce error by factor of 10^6.

(2) <u>Diffusion Equation</u>:
b = 1, $\rho = 0.11$, 6-7 iterations to reduce error by factor of 10^6.
b = 0.1, $\rho = 0.05$, 4-5 iterations to reduce error by factor of 10^6.
b = 0.01, $\rho = 0.035$, 4 iterations to reduce error by factor of 10^6.

There are also other methods to approximately solve

$$(I - \frac{b\Delta t}{2}L)u^{n+1} = (I + \frac{b\Delta t}{2}L)u^n$$

directly (i.e. approximately invert matrix without iterating). The work required is $O(M) = O(N^2)$. These methods are based on the fact that we are timestepping, and are thus not available for the poisson equation. There are two basic methods:

1. ADI Methods (Alternating Direction Implicit)
2. LOD Methods (Locally 1-dimensional)

Both of these are examples of operator splitting (or fractional step) methods. We split the Laplacian operator as $\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial x^2} = L_x + L_y$.

LOD Method:
Here Essentially we diffuse only in the x direction first and then only in the y direction. This gives a two step process:

$$(I - \frac{b\Delta t}{2} L_x)u^* = (I + \frac{b\Delta t}{2} L_x)u^n$$

$$(I - \frac{b\Delta t}{2} L_y)u^{n+1} = (I + \frac{b\Delta t}{2} L_y)u^*$$

ADI Method:
This is similiar to the LOD method but we instead couple the x and y components in each 1/2 timestep. It turns out to work better:

$$(I - \frac{b\Delta t}{2} L_x)u^* = (I + \frac{b\Delta t}{2} L_y)u^n$$

$$(I - \frac{b\Delta t}{2} L_y)u^{n+1} = (I + \frac{b\Delta t}{2} L_x)u^*$$

work estimate:
Diffusion in x $\rightarrow$ Solve N_y tridiagonal systems
Diffusion in y $\rightarrow$ Solve N_x tridiagonal systems
Each of N_y tridiagonal systems is size N_x
Each of N_x tridiagonal systems is size N_y
Total work is $O(N_x N_y + N_y N_x) = O(M)$

Stability:
We perform Von Neuman Analysis for the LOD Method.

$$\hat{u^*} = \left(\frac{1 - (4b\Delta t/h^2)\sin^2(\xi_1 h/2)}{1 + (4b\Delta t/h^2)\sin^2(\xi_1 h/2)} \right) \hat{u^n}$$

$$\hat{u^{n+1}} = \left(\frac{1 - (4b\Delta t/h^2)\sin^2(\xi_2 h/2)}{1 + (4b\Delta t/h^2)\sin^2(\xi_2 h/2)} \right) \hat{u^*}$$

combining these two equations gives:

$$\hat{u^{n+1}} = \left(\frac{(1-(4b\Delta t/h^2)\sin^2(\xi_1 h/2))(1-(4b\Delta t/h^2)\sin^2(\xi_2 h/2))}{(1+(4b\Delta t/h^2)\sin^2(\xi_1 h/2))(1+(4b\Delta t/h^2)\sin^2(\xi_2 h/2))}\right)\hat{u^n} = \rho(\xi_1,\xi_2)\hat{u^n}$$

where $\rho(\xi_1, \xi_2) \leq 1$. Hence, the method is (unconditionally) stable. The result $\hat{u^{n+1}} = \rho(\xi_1, \xi_2)\hat{u^n}$ is exactly the same for the ADI Method, so ADI is stable as well.

Consistency:
Showing consistency of LOD, ADI is actually harder than showing stability. We will show consistency only for ADI. To use the same proof for LOD we must assume L_x and L_y commute. This is true for periodic BC and constant coefficients but NOT in general.

To show consistency for ADI we apply $(I + \frac{b\Delta t}{2}L_x)$ to both sides of the equation for the first half time step in ADI and use that $(I - \frac{b\Delta t}{2}L_x)$ and $(I + \frac{b\Delta t}{2}L_x)$ commute:

$$(I-(b\Delta t/2)L_x)u^* = (I+(b\Delta t/2)L_y)u^n$$
$$(I+(b\Delta t/2)L_x)(I-(b\Delta t/2)L_x)u^* = (I+(b\Delta t/2)L_x)(I+(b\Delta t/2)L_y)u^n$$
$$(I-(b\Delta t/2)L_x)(I+(b\Delta t/2)L_x)u^* = (I+(b\Delta t/2)L_x)(I+(b\Delta t/2)L_y)u^n$$
$$(I-(b\Delta t/2)L_x)(I-(b\Delta t/2)L_y)u^{n+1} = (I+(b\Delta t/2)L_x)(I+(b\Delta t/2)L_y)u^n$$
$$(I-(b\Delta t/2)L_x-(b\Delta t/2)L_y+(b\Delta t/2)^2L_xL_y)u^{n+1} = (I+(b\Delta t/2)L_x+(b\Delta t/2)L_y+(b\Delta t/2)^2L_xL_y)u^n$$
$$(I-(b\Delta t/2)Lu^{n+1} = (I+(b\Delta t/2)Lu^n - (b\Delta t/2)^2L_xL_y)(u^{n+1}-u^n)$$
$$\frac{u^{n+1}-u^n}{\Delta t} = (b/2)(Lu^{n+1}+Lu^n) - (b\Delta t/2)^2L_xL_y(u^{n+1}-u^n)$$

So we get the Regular Crank-Nicolosn discretization error and an additional error term $O(\Delta t^2)$. Overall the scheme is still 2nd order accurate in Δt and h.

Note: We have shown ADI is stable and consistent so by the Lax-Equivalence theorem it is also convergent.