Lecture 19

Tuesday, December 4, 2007

Supplementary Reading: Osher and Fedkiw, §18.3, §23.1

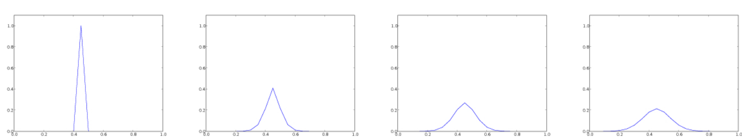
1 Heat Equation

The Heat Equation $\frac{\partial T}{\partial t} = \nabla \cdot (k\nabla T)$ is our model parabolic equation, and arises in several physical simulation applications.

1. We are commonly given initial values for T as $T(x, t = 0) = T^0(x)$ and boundary conditions for $t > 0$.

2. In the case where the spatial component of T is one-dimensional, we have the equation $T_t = (kT_x)_x$.

3. Parabolic equations approach a **steady state**. For example, in the heat equation we can take $T_t = 0$, which gives $\nabla \cdot (k\nabla T) = 0$, the Laplace equation.



Derivation

Starting from conservation of mass, momentum and energy one can derive

$$\rho e_t + \rho \vec{V} \cdot \nabla e + p\nabla \cdot \vec{V} = \nabla \cdot (k\nabla T) \tag{1}$$

where

k:	thermal conductivity
T:	temperature
e:	internal energy/unit mass
ρe:	internal energy/unit volume

Ideal Material and Divergence-Free

We first make the ideal material assumption, ie. e and T satisfy the relationship

$$de = c_v dT$$

and that our domain is divergence-free ($\nabla \cdot \vec{V} = 0$) to simplify equation 1 to

$$\rho c_v T_t + \rho c_v \vec{V} \cdot \nabla T = \nabla \cdot (k \nabla T) \tag{2}$$

which can be further simplified to the standard heat equation

$$\rho c_v T_t = \nabla \cdot (k \nabla T) \tag{3}$$

by ignoring the effects of convection, i.e. setting $\vec{V} = 0$. If k is constant, this can be written as

$$T_t = \frac{k}{\rho c_v} \triangle T. \tag{4}$$

Discretization

Applying explicit Euler time discretization to equation 3 results in

$$\frac{T^{n+1} - T^n}{\triangle t} = \frac{1}{\rho c_v} \nabla \cdot (k \nabla T^n) \tag{5}$$

where either Dirichlet or Neumann boundary conditions can be applied on the boundaries of the computational domain. Assuming that ρ and c_v are constants allows us to rewrite this equation as

$$\frac{T^{n+1} - T^n}{\triangle t} = \nabla \cdot \left(\hat{k} \nabla T^n \right) \tag{6}$$

with $\hat{k} = \frac{k}{\rho c_v}$. Standard central differencing (second order accurate) can be used for the spatial derivatives as in

$$\frac{\hat{k}_{i+\frac{1}{2},j} \left(\frac{T_{i+1,j} - T_{i,j}}{\triangle x} \right) - \hat{k}_{i-\frac{1}{2},j} \left(\frac{T_{i,j} - T_{i-1,j}}{\triangle x} \right)}{\triangle x}$$

A time step restriction of

$$\triangle t \hat{k} \left(\frac{2}{(\triangle x)^2} + \frac{2}{(\triangle y)^2} + \frac{2}{(\triangle z)^2} \right) \leq 1 \tag{7}$$

is needed for stability. If we $\triangle x = \triangle y$, then this is

$$4 \frac{\triangle t}{\triangle x^2} \hat{k} \leq 1$$

In $3D$, the restriction is $6\frac{\triangle t}{\triangle x^2}\hat{k} \leq 1$ and in general for nD the restriction is $2n\frac{\triangle t}{\triangle x^2}\hat{k} \leq 1$.

Implicit Euler time discretization

$$\frac{T^{n+1} - T^n}{\triangle t} = \nabla \cdot \left(\hat{k}\nabla T^{n+1}\right) \tag{8}$$

avoids this time step stability restriction. This equation can be rewritten as

$$T^{n+1} - \triangle t \nabla \cdot \left(\hat{k}\nabla T^{n+1}\right) = T^n \tag{9}$$

discretizing the $\nabla \cdot \left(\hat{k}\nabla T^{n+1}\right)$ term using central differencing. For each unknown, T_i^{n+1}, equation 9 is used to fill in one row of a matrix creating a linear system of equations. Since the resulting matrix is symmetric, a number of fast linear solvers can be used (e.g. a PCG method with an incomplete Choleski preconditioner, see Golub and Van Loan [1]). Equation 8 is first order accurate in time and second order accurate in space, and $\triangle t$ needs to be chosen proportional to $\triangle x^2$ in order to obtain an overall asymptotic accuracy of $O(\triangle x^2)$. However, the stability of the implicit Euler method allows one to chose $\triangle t$ proportional to $\triangle x$ saving dramatically on CPU time. The Crank-Nicolson scheme

$$\frac{T^{n+1} - T^n}{\triangle t} = \frac{1}{2}\nabla \cdot \left(\hat{k}\nabla T^{n+1}\right) + \frac{1}{2}\nabla \cdot \left(\hat{k}\nabla T^{n}\right) \tag{10}$$

can be used to achieve second order accuracy in both space and time with $\triangle t$ proportional to $\triangle x$. For the Crank-Nicolson scheme,

$$T^{n+1} - \frac{\triangle t}{2}\nabla \cdot \left(\hat{k}\nabla T^{n+1}\right) = T^n + \frac{\triangle t}{2}\nabla \cdot \left(\hat{k}\nabla T^{n}\right) \tag{11}$$

is used to create a symmetric linear system of equations for the unknowns T_i^{n+1}. Again, all spatial derivatives are computed using standard central differencing.

Why not always use Crank-Nicholson, as it gives second order accuracy and no time step restriction? Let us look at the solution as $\triangle t \to \infty$. Backward Euler gives

$$\triangle T^n = 0,$$

which is the correct steady state solution. Crank-Nicholson gives

$$\triangle T^{n+1} = -\triangle T^n.$$

In $1D$ this is

$$T_{xx}^{n+1} = -T_{xx}^n$$

This shows that the curvature is changing sign at each time step. So the problem with Crank-Nicholson is that as $\triangle t$ gets very large, you get oscillations, whereas with backward Euler, you get the steady-state solution.

References

[1] Golub, G. and Van Loan, C., *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 1989.