

The Finite Difference Method for Parabolic Problems

Varun Shankar

March 1, 2016

1 Introduction

In the previous chapter, we studied the application of the FDM to Poisson's equation, a purely elliptic PDE. Now, we will discuss its application to the heat equation, a parabolic problem.

Specifically, we will focus on the heat equation with Dirichlet boundary conditions:

$$\frac{\partial u}{\partial t} = D\Delta u, \mathbf{x} \in \Omega, \quad (1)$$

$$u(\mathbf{x}) = g(\mathbf{x}), \mathbf{x} \in \partial\Omega. \quad (2)$$

This time, we will jump directly to 2D, and leave out any discussion of Neumann boundary conditions. Everything from Poisson's equation carries over here naturally.

2 Method of Lines

Our primary approach to solving space-time PDEs will be the Method of Lines approach. Basically, this approach boils down to the following two steps:

1. First discretize any spatial differential operators. This converts the PDE into a large system of ODEs, with one ODE at each grid point.
2. Now, pick a time integrator from the ODE chapter: Adams, BDF or RK methods (or some other method of your choice). Advance the ODEs forward in time.

In the case of the heat equation, imagine we discretize the Laplacian with the appropriate boundary conditions. Recall that we call the discrete Laplacian L . Let $\mathbf{U}$ be the vector of unknowns on the grid, ordered lexicographically (or just

consistently with the ordering in L). Then, we now have a system of ODEs of the form

$$\frac{d\mathbf{U}}{dt} = D\mathbf{L}\mathbf{U}. \quad (3)$$

This is a fairly straightforward system of ODEs: from the ODE perspective, L is a linear operator, making the right hand side very simple. Of course, we must consider the *properties* of this system of ODEs. Comparing this to the test ODE $y' = \lambda y$ that we used to calculate stability, we must now ask ourselves whether this system of ODEs is *stiff*. If it *is* stiff, we should select our time integrator carefully so as to avoid time-step restrictions.

To see whether this ODE is stiff, we must ask ourselves about the “magnitudes” associated with L . We know that the “magnitude” of a matrix is, in the most basic form, encoded in its set of eigenvalues. Then, the stiffness of each ODE in the system of ODEs is dictated primarily by the eigenvalues of L . What do we know about the eigenvalues of L ? Well, since L is symmetric negative-definite for Dirichlet BCs, and symmetric negative semi-definite (one zero eigenvalue) for Neumann BCs, we know L has negative eigenvalues, much like its continuous counterparts. Further, looking at its continuous counterpart Δ , we realize that Δ has an infinite number of eigenvalues, leading to a very large spread. While L only has a finite number of eigenvalues, they will also be spread quite far apart. This fits our definition of stiffness. Thus, the system of ODEs is stiff. We must turn to implicit time integrators. We will now explore four time integrators: forward Euler (explicit, but simple), Backward Euler (implicit, first order in time), Crank-Nicolson (aka AM2 aka the Trapezoidal rule, implicit, second order in time), and BDF2 (implicit, second order in time).

2.1 Forward Euler for the Heat Equation

We already know this is a bad idea. However, FE will serve as a template for understanding how to solve the above system of ODEs with more complicated time integrators. Proceeding with the FE discretization, we have

$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} = D\mathbf{L}\mathbf{U}^n. \quad (4)$$

Forward Euler, obviously, will impose a *severe* time-step restriction. We will see how severe when analyzing the stability of these schemes. For now, we will simply remark that using FE for the heat equation is a bad idea.

2.2 Backward Euler for the Heat Equation

Backward Euler is an implicit, first-order method. It can be viewed as both AM1 and BDF1. As we have already discussed, it is an A-stable method. This

means that regardless of the spread of the eigenvalues of L , backward Euler will allow us to take large time-steps and still obtain stability. The time-step size is now purely selected for reasons of accuracy, and in practice, we will pick as large a time-step as we can get away with. The BE discretization looks like:

$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} = D L \mathbf{U}^{n+1}, \quad (5)$$

$$\mathbf{U}^{n+1} - D \Delta t L \mathbf{U}^{n+1} = \mathbf{U}^n, \quad (6)$$

$$(I - D \Delta t L) \mathbf{U}^{n+1} = \mathbf{U}^n. \quad (7)$$

Recall that not only is BE A-stable, it also possesses *stiff decay*. This means that the solution modes will dissipate as rapidly as the true solution to the heat equation— a desirable property indeed! To solve for $\mathbf{U}^{n+1}$, we need to invert the large, *sparse* matrix on the left hand side. We already know how to do this for different properties of the matrix from our discussion on iterative methods.

2.3 Crank-Nicolson for the Heat Equation

CN is an extremely popular scheme for the heat equation. Its popularity is due to the fact that it allows second order accuracy using a *single step*; this is because it is an Adams-Moulton method (contrast to the AB or BDF methods). Further, recall that the trapezoidal rule (aka CN aka AM2) has the smallest truncation error of all second-order schemes. Even more importantly, recall that CN is A-stable! Clearly, it is a fantastic method. The CN discretization looks like:

$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} = \frac{D}{2} (L \mathbf{U}^{n+1} + L \mathbf{U}^n), \quad (8)$$

$$\mathbf{U}^{n+1} - \frac{D \Delta t}{2} L \mathbf{U}^{n+1} = \mathbf{U}^n + \frac{D \Delta t}{2} L \mathbf{U}^n, \quad (9)$$

$$\left(I - \frac{D \Delta t}{2} L \right) \mathbf{U}^{n+1} = \left(I + \frac{D \Delta t}{2} L \right) \mathbf{U}^n. \quad (10)$$

Since CN is A-stable, it is unconditionally stable on the heat equation, just like BE, but it offers $O(\Delta t^2)$ convergence. Looking at it more carefully, we realize that we have an extra matrix-vector multiplication on the right hand side. Once again, we must invert a large, *sparse* matrix to find $\mathbf{U}^{n+1}$.

Often lost in the discussion of Crank-Nicolson is its one important drawback: it does not possess stiff-decay. Indeed, the trapezoidal rule is a symplectic integrator, which means that it will not add any dissipation of its own to the solution of the PDE. This means that CN may cause certain solution modes to decay more slowly than the solution to the heat equation itself. For certain kinds of boundary conditions, this may manifest as a reduction in order of convergence unless Δt is chosen sufficiently small. This leads us to our final time integration scheme.

2.4 BDF2 for the Heat Equation

The BDF2 method, if you recall, is A-stable, second-order accurate, and possesses *stiff decay*. However, it requires storage of the solution from time t_{n-1} , since unlike AM2, it is a two-step scheme. The BDF2 scheme looks like this:

$$\left(I - \frac{2}{3}D\Delta tL\right)\mathbf{U}^{n+1} = \frac{4}{3}\mathbf{U}^n - \frac{1}{3}\mathbf{U}^{n-1}. \quad (11)$$

The stiff decay property of BDF2 makes it more “stable” than Crank-Nicolson across a wider range of parameters, in the sense discussed earlier. Further, it is more computationally efficient than Crank-Nicolson, requiring one less matrix-vector multiply per time-step.

2.5 Other methods

One could certainly use *any* of the implicit Runge-Kutta (IRK) methods to solve the heat equation. They all possess both A-stability and L-stability (the RK version of stiff decay). However, the cost of RK methods increases faster based on the number of ODEs than the cost of implicit multistep methods. For order 2 or less, it is hard to beat an LMM for the heat equation. Further, in practice, since BDF3 and BDF4 are almost A-stable, it is quite safe to use them for higher-order time integration of the heat equation (in conjunction with higher-order spatial discretizations).