Computational Methods for Financial Applications, Lecture Notes

Jonathan Goodman
Courant Institute of Mathematical Science, NYU

September 17, 1996

1 Finite Difference Methods for Initial Value Problems

Before discussing specific problems, I want to avoid later confusion by giving two general definitions.

Well Posed Problem: A computation generally seeks to approximate the solution to some mathematically posed problem. Generally, the statement of a problem has two parts.

Conditions: a Partial Differential Equation (PDE), boundary conditions, constraints, and so on. The conditions remain the same from one instance of a problem to another.

Data: initial values, final values (e.g. a payout function), boundary values, and so on. These distinguish one instance of a problem from another.

The *solution* to the problem will be called u in this general discussion. The data will be called f. A problem is called *well posed* if:

Uniqueness. The solution, u, is completely determined by the data, f, and the conditions.

Continuity. The solution depends continuously on the data in some suitable sense.

These definitions are not completely precise. For example, it is not completely clear what parts of a problem formulation should be classified as conditions and what parts as data. Changes in this classification could effect whether a problem is determined to be well posed or not. I will take the point of view that an ill posed problem (one that is not well posed) is computationally intractable and argue that it is a mistake to attempt a numerical solution of an ill posed problem. A more precise definition of well posedness was given by the French turn of the century mathematician Hadamard, I believe, in connection with his study of stability and fracture of elastic materials.

Example. Black and Scholes (and Samuelson before them) proposed to compute the present value of a stock option as a function of the current price of the stock, S, by solving the Black-Scholes partial differential equation:

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} = rf \quad , \tag{1}$$

where r and σ are parameters. For a European option, the conditions consist completely of this PDE. The data consists of the value, $f(S,T)$, of the option at the time it expires, the "payout". The solution is the value of the option at the current time, $f(S,0)$. Unfortunately, what was called u in the general discussion above is called f here. It will prove impossible to be more consistent with notation. Turning the problem around, specifying the present value of the option and trying to determine a final time payout function, $F(S,T)$, that gives the specified present value function, leads to an ill posed problem. It is in general impossible to "engineer" an arbitrary present value function by picking the future payout. I hope the reason for this will become clearer later.

Linear. A problem class is linear if it satisfies the *superposition principle.* In quantum mechanics the superposition principle is thought to be a fundamental physical fact (hence the term principle). Here, the superposition principle is not a principle, but the name of a property that is true for some problems and not for others. The superposition principle holds for a problem class if that class has the following property: Suppose data f_1 leads to solution u_1 and data f_2 leads to solution u_2, then data $f_1 + f_2$ leads to solution $u_1 + u_2$. Also, data $C \cdot f_1$ leads to solution $C \cdot u_1$. In other words, a problem is linear if it satisfies the rules of linear algebra. If we know a problem is linear, then we can construct lots of solutions by superposing known solutions. The Black

Scholes formula for the solution of (1) for European puts and calls is found in this way.

Example. The valuation problem for American options is slightly different from (1). The conditions are roughly as follows: There is an "intrinsic value", $f_{\text{int}}(S)$ of an option. The value of the option satisfies (1) only when it is larger than the intrinsic value. The option is never worth less than the intrinsic value. The American option valuation problem does not satisfy the superposition principle. It is rare that the sum of two solutions is again a solution. We often say that while the PDE (1) is linear, the boundary condition, an inequality involving the intrinsic value, is nonlinear. This example shows that a problem may be nonlinear even when the PDE is linear.

Model problem. A model problem is a computational problem that is much simpler than the problem at had that helps us to understand the problem we are actually interested in. Using model problems to elucidate phenomena (perhaps only in a qualitative way) in complex problems had been a fruitful tradition in applied mathematics. We often choose model problems we already know the answer to in order to study errors produced by numerical approximation methods. When we develop methods for model problems, we must remember to consider only methods that also apply to the "real" problems we are actually interested in.

1.1 Finite Difference Marching Methods for the Heat Equation

The heat equation in one dimension is a very simple model for general parabolic evolution equations such as (1) that arise in financial applications. The problem is as follows: we seek to find a function of x (the space variable) and t (the time variable) that satisfies the PDE (for $t > 0$)

$$\frac{\partial u}{\partial t} = \frac{1}{2}\frac{\partial^2 u}{\partial t^2} \quad , \tag{2}$$

together with initial condition

$$u(x,0) = f(x) \quad , \tag{3}$$

where $f(x)$ is some given function. We consider finite differency methods on a grid with "space step" Δx and "time step" Δt. The grid is the set of lattice points in space time (x_j, t_n) where $x_j = j\Delta x$ and $t_n = n\Delta t$. Here, n and j are integers. The marching method will compute approximations to the values of u on the grid, namely

$$U_j^n \approx u(x_j, t_n) \quad . \tag{4}$$

For the next few paragraphs, we will consider the idealized situation where all integers j are allowed. Since this is an infinite number, none of these algorithms can be programmed exactly on a computer. Nevertheless, the analysis will be very helpful and can be adapted to the more realistic situation where there are some kind of boundary conditions that restrict the range of the j variable.

A marching method computes U values for successively increasing n (that is, by "marching"). First, the values U_j^0 are taken to be the given initial data:

$$U_j^0 = f(x_j) \quad .$$

Now suppose we have computed the U_j^n for all j values. We want to use a finite difference approximation to the equation (2) to compute the U_j^{n+1} values for all j. The simplest such method is the "forward Euler" method, which we now describe. If we approximate the partial derivatives in (2) by finite differences using the formulae:

$$\frac{\partial u}{\partial t} \approx \frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} \quad , \tag{5}$$

and

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{\Delta x^2} \quad , \tag{6}$$

we get the approximate relation

$$u(x_j, t_{n+1}) \approx u(x_j, t_n) + \frac{\Delta t}{\Delta x^2} \left(u(x_{j+1}, t_n) - 2u(x_j, t_n) + u(x_{j-1}, t_n) \right) \quad . \tag{7}$$

The forward Euler approximate solution is gotten by taking this approximate relation as the exact definition of the approximate solution, U:

$$U_j^{n+1} = U_j^n + \frac{\Delta t}{\Delta x^2} \left(U_{j+1}^n - 2U_j^n + U_{j-1}^n \right) ll. \tag{8}$$

An approximation such as (8) is said to *converge* if

$$U_j^n - u(x_j, t_n) \to 0 \;\text{ as } \Delta x \to 0 \text{ and } \Delta t \to 0 \text{ for all } t_n \leq T.$$

Clearly, the most important question about an approximation is whether it converges. Once we know an approximation converges, we can ask for useful error estimates or bounds, qualitative properties such as oscillations, etc.

The fundamental concepts in the convergence analysis of finite difference methods are *consistency* and *stability*. Roughly, a method is consistent if its *truncation error* goes to zero as $\Delta x \to 0$ and $\Delta t \to 0$. This condition is generally easy to check using Taylor series analysis of the difference approximations such as (4) and (5). A scheme is stable if errors introduced at early stages of the computation are not amplified too much in later stages. The fundamental fact is

Theorem. (Lax) *If a marching method is consistent and stable, then it converges.*

Consistency and *truncation error* are defined for any finite difference method. To postpone an unmotivated general technical discussion, I want to focus on truncation error for the forward Euler method. Truncation error, ρ, is (up to a factor of Δt) the amount by which the exact solution, u, fails to satisfy the finite difference equation (8). That is,

$$u(x_j, t_{n+1}) = u(x_j, t_n) + \frac{\Delta t}{\Delta x^2}\left(u(x_{j+1}, t_n) - 2u(x_j, t_n) + u(x_{j-1}, t_n)\right) + \Delta t \rho_j^n \ . \tag{9}$$

The forward Euler method is consistent because, if u is a smooth function of x and t, then

$$m(t, \Delta x, \Delta t) = \max_{j, t_n \le t} \left|\rho_j^n\right| \to 0 \quad \text{as } \Delta t \to 0 \text{ and } \Delta x \to 0.$$

Since (7) is a combination of (5) and (6), the total truncation error is the sum of the truncation error due to (5) and that due to (6). That is $\rho_j^n = \sigma_j^n + \tau_j^n$, where

$$\sigma_j^n = \partial_t u(x_j, t_n) - \frac{u(x_j, t_n + \Delta t) - u(x_j, t_n)}{\Delta t} \ ,$$

and

$$\tau_j^n = \partial_x^2 u(x_j, t_n) - \frac{u(x_j + \Delta x, t_n) - 2u(x_j, t_n) + u(x_j - \Delta x, t_n)}{\Delta x^2} \ .$$

Taylor series approximations that are valid when u is smooth and Δx and Δt are small lead to

$$\sigma_j^n \approx \frac{1}{2}\Delta t \partial_t^2 u(x_j, t_n) \ ,$$

and

$$\tau_j^n \approx \frac{1}{12}\Delta x^2 \partial_x^4 u(x_j, t_n) \quad ,$$

so

$$\rho_j^n \approx \frac{1}{2}\Delta t \partial_t^2 u(x_j, t_n) + \frac{1}{12}\Delta x^2 \partial_x^4 u(x_j, t_n)$$

does indeed go to zero as $\Delta x \to 0$ and $\Delta t \to 0$.

Even a consistent approximation method may fail to converge if it is not stable. Since stability is about amplification of errors, it is defined (for linear problems such as our model problem) in terms of norms. Thus, let $\|U^n\|$ be a vector space norm such as

$$\|U^n\|_{l^2} = \sqrt{\sum_j \left(U_j^n\right)^2} \quad ,$$

$$\|U^n\|_{l^1} = \sum_j \left|U_j^n\right| \quad ,$$

$$\|U^n\|_{l^\infty} = \sup_j \left|U_j^n\right| \quad .$$

A scheme is stable if there is a constant independent of Δx, Δt, n, and U^0 so that

$$\|U^n\| \leq C \left\|U^0\right\| \quad . \tag{10}$$

Unfortunately, it is possible for a scheme to be stable in one norm (say, $\|\cdot\|_{l^2}$) while being unstable in another norm (say, $\|\cdot\|_{l^\infty}$). This is so uncommon for parabolic equations that we will ignore this possibility from now on.