Computational Methods in Finance, Lecture 2, Diffusions and Diffusion Equations.

Jonathan Goodman *
Courant Institute of Mathematical Sciences, NYU

October 1, 1997

1 Introduction

This lecture and the next are about finite difference methods for solving the diffusion equations that arise in financial applications. Before get to finite differences, we review stochastic differential equations. As in Lecture 1, we discuss the forward and backward equations and the differences between them.

2 Diffusions and Diffusion Equations

Lecture 1 discusses Markov processes in discrete state space and discrete time. Now we turn to continuous time and continuous state space. The state at time t is a vector, $X \in \mathbf{R}^n$ consisting of n components, or "factors", $X = (X_1, \ldots, X_n)$. The dynamics are given by the Ito differential equation

$$dX(t) = a(X(t))dt + b(X(t))dZ \quad . \tag{1}$$

Here $Z(t)$ is a vector of m independent standard Brownian motions. For each x, there is a "drift", $a(x)$, and an $n \times m$ matrix $b(x)$, that is related

*goodman@cims.nyu.edu, or http://www.math.nyu.edu/faculty/goodman,

to the volatility. There is no reason that m, the number of noise sources, should equal n, the number of factors, but there is no reason ever to have more noises than factors. The columns of b are vectors in $\mathbf{R}^n$. Column j gives the influence of noise j on the dynamics. If these columns do not span $\mathbf{R}^n$, then the diffusion is "degenerate". Otherwise, it is nondegenerate. Both types come up in financial applications.

As in Lecture 1, there are forward and backward evolution equations that are dual to each other. The forward equation is for $u(x,t)$, the probability density for $X(t)$. This is the "diffusion equation"

$$\partial_t u = -\sum_{j=1}^{n} \partial_{x_j} a_j(x) u + \frac{1}{2} \sum_{j,k=1}^{n} \partial_{x_j} \partial_{x_k} \mu_{jk}(x) u \quad . \tag{2}$$

The matrix of diffusion coefficients, μ, is related to b by

$$\mu(x) = b(x) \cdot b^*(x) \quad . \tag{3}$$

We write M^* for the transpose of a matrix, M. The coefficients, $a_j(x)$, in (2) are the components of a in (1).

A strict mathematical derivation of (2) from (1) or vice versa is outside the scope of this course. However, some aspects of (2) that are natural. First, because u is a probability density, the integral of $u(x,t)$ over x should be independent of t. That will happen if all the terms on the right of (2) are derivatives of something. This is sometimes called "conservation form". The second term on the right of (2) involves two derivatives. Someone used to physical modeling might expect it to take the form

$$\text{WRONG} \qquad \frac{1}{2} \sum_{j,k=1}^{n} \partial_{x_j} \mu_{jk}(x) \partial_{x_k} u \quad . \qquad \text{WRONG} \tag{4}$$

The actual form (2) has the "martingale" property that, if there is no drift, then the expected value of X does not change with time. To see this, use (2) with $a \equiv 0$ and compute

$$\begin{aligned} \partial_t \mathbf{E}\left[X(t)\right] &= \partial_t \int x u(x,t) dx \\ &= \int x \frac{1}{2} \sum_{j,k=1}^{n} \partial_{x_j} \partial_{x_k} \mu_{jk}(x) u d \\ &= 0 \ . \end{aligned}$$

The last line follow from the one above if you integrate by parts twice to put the two derivatives on the x. The result would generally not be zero using (4) instead of (2).

Here is an equally rigorous derivation of the relation (3) between volatility and diffusion coefficients. Suppose first that $n = m = 1$. The number μ should depend on b in some way. Observe that the diffusion governed by (1) will be unchanged of b is replaced by $-b$; $Z(t)$ is indistinguishable from $-Z(t)$. This suggests the formula $\mu = b^2$. In general, we need a matrix analogue of $\mu = b^2$ that produces an $n \times n$ matrix, μ, from an $n \times m$ matrix, b. The simplest possibility is (3). For constant b and $a = 0$, we can match moments. Use (1) and assume $X(0) = 0$ and $Z(0) = 0$[1] and get $X(t) = bZ(t)$. From this it follows that

$$\mathrm{cov}\left(X(t)X^*(t)\right) = bb^*t \quad .$$

On the other hand, from (2) we compute

$$\begin{aligned} \partial_t \mathbf{E}\left(XX^*\right) &= \partial_t \int xx^* u(x,t)dx \\ &= \frac{1}{2}\int xx^* \sum_{jk} \partial_{x_j}\partial_{x_k}\mu_{jk} u dx \\ &= \mu \quad . \end{aligned}$$

which again agrees with (3).

The drift term in (2), $\partial_x au$, corresponds to the drift term in (1), $a(X)dt$. It is easy to see what the term would be if a were constant (independent of x and t) and b were zero. In that case the solution of (1) would be $X(t) = X(0) + at$. For this reason, the probability density is also simply shifted with speed a: $u(x,t) = u(x - at, 0)$. This function satisfies (2) (if the signs are right).

As in Lecture 1, the simplest backward equation is for

$$f(x,t) = \mathbf{E}\left[f_T(X(T)) | X(t) = x\right] \quad .$$

More complicated expectations satisfy more complicated but related equations. The backward equation for f is

$$\partial_t f + \sum_j a_j(x)\partial_{x_j} f + \frac{1}{2}\sum_{jk}\mu_{jk}(x)\partial_{x_j}\partial_{x_k} f = 0 \quad . \tag{5}$$

[1] The Wiener process is usually defined to that $Z(0) = 0$.

Still following Lecture 1, this is supplemented with "initial data" given at the final time, T, $f(x,T) = f_T(x)$, and determines $f(x,t)$ for $t < T$. Again, we can express unconditional expectation in terms of conditional expectation starting from time t and the probability density for $X(t)$:

$$\mathbf{E}\left[f_T(X(T))\right] = \int f(x,t)u(x,t)dx \quad . \tag{6}$$

The fact that the right side of (6) is independent of t allows us to derive (5) from (2) or vice versa. Finally, f satisfies a "maximum principle":

$$\min_y f(y,T) \leq f(x,t) \leq \max_y f(y,T) \quad \text{if } t < T.$$

The probability interpretation of f makes this obvious; the expected reward cannot be less than the least possible reward nor larger than the largest.

In addition to initial conditions, diffusion equations often come with boundary conditions. In this case, the "domain" will be a subset of all possible x values. The boundary of the domain will be called B. Financial applications often give rise to diffusion equations without boundary conditions. For example, there are no natural restrictions on the price of a stock. Boundary conditions do come in, for example, when working with knockout features. A knockout at a set B means that there will be no payout if $X(t) \in B$ for any $0 \leq t \leq T$. To value such an instrument, we use the backwards equation with "Dirichlet" boundary condition $f(x,t) = 0$ for $x \in B$. If the forward equation is applicable, we also apply the Dirichlet boundary condition to u. In the latter case, we usually have $\int u dx < 1$. This $u(x,t)$ represents the probability density for those paths that have never touched the boundary. The complimentary probability is the probability of touching the boundary at some time:

$$\int u(x,t)dx + \mathbf{Pr}\left(X(t') \in B \text{ for some } t' \leq T\right) = 1 \quad .$$

Most diffusions in finance do not "live" in all of R^n, but in a natural subset. For example, stock prices and interest rates are usually positive. In these cases, the diffusion coefficients may go to zero, as X gets close to the edge, in such a way that $X(t)$ can never leave the set. For example, in the model used by Black and Scholes, $dS = rSdt + \sigma SdZ$, the $S(t)$ can never become negative if it starts positive. In these cases, no extra boundary conditions need to be specified. We do not give a boundary condition at $S = 0$ when solving the Black Scholes equation.

Often the initial data, $f(x, T)$, or $u(x, 0)$, are singular. A singularity is an x value where the function is not smooth. For example, if $X(0) = x_0$ is known, then the initial probability density is a delta function: $u(x, 0) = \delta(x - x_0)$. The payout for a stock option has a jump in its derivative at the strike price. If the diffusion is nondegenerate (the coefficient matrix, μ, is positive definite), such singularities quickly smooth out. This may or may not be true for degenerate diffusions.

3 A Model Problem

In designing and understanding computational methods, we try to identify model problems. A model problem is a simpler computation that helps us focus on the essential difficulties while temporarily avoiding other complexities of our actual problem. We must keep in mind that a model problem can be an oversimplification, some of its features are not shared by the actual problem.

Our first model problem for diffusion equations will be the "heat equation"

$$\partial_t u = \frac{1}{2} \partial_x^2 u \tag{7}$$

in one space dimension. This is to be solved together with initial data $u(x, 0) = u_0(x)$. This problem has in common with general diffusion equations (2) or (3) the smoothing property, time step constraints for explicit difference methods, and infinite propagation speed with limited spreading. It has the unusual features that the forward and backward equations are (up to a sign) the same, and that there is a simple algebraic formula for the fundamental solution.

4 Finite differences for the heat equation

Finite differences are approximations of derivatives. For example, there is the one sided approximation to the first derivative,

$$\frac{dg}{ds} \approx \frac{g(s + h) - g(s)}{h} \quad , \tag{8}$$

the centered approximation,

$$\frac{dg}{ds} \approx \frac{g(s+h) - g(s-h)}{2h} \quad , \tag{9}$$

and the three point approximation to the second derivative,

$$\frac{d^2g}{ds^2} \approx \frac{g(s+h) - 2g(s) + g(s-h)}{h^2} \quad . \tag{10}$$

Here, h is a small "step". If g is a smooth function of s then each of the approximations improves as h gets smaller.

The finite difference approximation to the solution of the diffusion equation consists of values, U_j^n, that are supposed to approximate $u(x_j, t_n)$. We use a small "space step", Δx, and a "time step", Δt. The grid points, (x_j, t_n), are given by $x_j = j\Delta x$ and $t_n = n\Delta t$. The grid points form a lattice in space and time with uniform spacing between them.

The idea is to compute new values, U_j^{n+1}, in terms of already known values, U_j^n. The formula for U_j^{n+1} is derived from (7) by inserting finite difference approximations for the derivatives. We use the one sided derivative (8) for the time derivative and the three point formula (10) for the x derivative. This gives (using $t_n + \Delta t = t_{n+1}$),

$$\begin{aligned} \partial_t u(x_j, t_n) &\approx \frac{u(x_j, t_n + \Delta t) - u(x_j, t_n)}{\Delta t} \\ &\approx \frac{U_j^{n+1} - U_j^n}{\Delta t} \quad , \end{aligned}$$

and

$$\begin{aligned} \partial_x^2 u(x_j, t_n) &\approx \frac{u(x_j + \Delta x, t_n) - 2u(x_j, t_n) + u(x_j - \Delta x, t_n)}{\Delta x^2} \\ &\approx \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{\Delta x^2} \quad . \end{aligned}$$

Inserting these into (7) gives

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{\Delta x^2} \quad ,$$

or

$$U_j^{n+1} = \frac{\Delta t}{2\Delta x^2} U_{j+1}^n + \left(1 - \frac{\Delta t}{\Delta x^2}\right) U_j^n + \frac{\Delta t}{2\Delta x^2} U_{j-1}^n \quad . \tag{11}$$

This is the forward Euler method for solving the heat equation. If we apply the same reasoning to the model backward equation,

$$\partial_t f + \frac{1}{2}\partial_x^2 f = 0 \ ,$$

we get

$$F_j^{n-1} = \frac{\Delta t}{2\Delta x^2} F_{j+1}^n + \left(1 - \frac{\Delta t}{\Delta x^2}\right) F_j^n + \frac{\Delta t}{2\Delta x^2} F_{j-1}^n \ . \tag{12}$$

This is also the forward Euler method because it too has time marching in the natural direction, in this case from the future towards the present.

The formula (11) allows us to compute the U_j^n using a "marching" algorithm. First, we take initial data for U to agree with the given initial data for u, $U_j^0 = u(x_j, 0)$ at the grid points[2] Then we use (11) with $n = 0$ to compute all the values U_j^1. We continue in this way until we have computed up to time $t_n = T$, our desired final time. If the final time is not exactly one of the t_n, we may have to interpolate to get values at time T. To solve the backward equation, we instead use (12) to march from given data at time T to the desired time $t = 0$. Note that we have used a different approximation to ∂_t than before. We will talk more about this in the discussion of implicit methods.

The marching method based on (11) will give an approximation to the correct answer only of a *time step constraint* is respected. In the case of the forward Euler method (11) or (12), the time constraint is satisfied if all the coefficients on the right side are positive. This leads to

$$\Delta t < \Delta x^2 \ . \tag{13}$$

This condition is sometimes called the CFL condition, after Courant (the founder of the Courant Institute), Friedrichs (who provided scientific leadership for the Institute for decades) and Lewy. The condition appears in a paper of theirs "Über die partiellen Differenzengleichungen der mathematischen Physik" published in 1928 ("gleichung" means equation).

We end with a note on programming. A large multidimensional computation may strain the memory of the computer. We can reduce the memory needed by keeping only the values U_j^{n+1} and U_j^n at any given stage of the

[2] If the initial data is not smooth, it may be wise to use a more sophisticated choice of the U_j^0.

computation. That is, we need two one dimensional arrays rather than a two dimensional array.