Derivative Securities, Fall 2010
Mathematics in Finance Program
Courant Institute of Mathematical Sciences, NYU
Jonathan Goodman
`http://www.math.nyu.edu/faculty/goodman/teaching/DerivSec10/resources.html`

Week 7

1 The backward equation

The last big piece of mathematical theory for this class is the relation between diffusions, value functions, and the partial differential equations they satisfy. We have already done so in the binomial model, the only difference being that there the backward is an algebraic equation not a differential equation. The relationship goes both ways. If we have a stochastic differential equation we can find partial differential equations associated to it. Conversely, it is possible to associate diffusion processes to certain partial differential equations, such as the Black Scholes equation.

Suppose X_t is a diffusion with drift coefficient $a(x,t)$ and infinitesimal variance $b^2(x,t)$. This means, as usual, that X_t is a continuous function of t and

$$E\left[dX_t \mid \mathcal{F}_t\right] \;=\; a(X_t,t)dt \;, \tag{1}$$

and

$$E\left[dX_t^2 \mid \mathcal{F}_t\right] \;=\; b^2(X_t,t)dt \;. \tag{2}$$

You also can think of X_t as the solution of the stochastic differential equation

$$dX_t \;=\; a(X)t,t)dt \,+\, b(X_t,t)dW_t \;. \tag{3}$$

In this discussion, the mean and variance formulas (1) (2) are more useful than the SDE (3).

There are several different kinds of value function with different associated PDEs, but here is the simplest. Suppose you get a payout $V(X_T)$ at time T. The value function, $f(x,t)$, is the expected value of the payout at time $t \leq T$ starting with $X_t = x$. This may be written

$$f(x,t) \;=\; E_{x,t}\left[V(X_T)\right] \;. \tag{4}$$

An equivalent way to say this is

$$f(X_t,t) \;=\; E\left[V(X_T) \mid \mathcal{F}_t\right] \;. \tag{5}$$

These are equivalent because $\mathcal{F}_t$ knows the value of X_t. For the same reason, (1) and (2) may be written as

$$E_{x,t}\left[dX_t\right] \;=\; a(x,t)dt \;, \tag{6}$$

and

$$E_{x,t}\left[dX_t^2\right] \;=\; b^2(x,t)dt \;. \tag{7}$$

Of course (4) or (5) imply that,

$$f(x,T) \;=\; V(x)\;, \tag{8}$$

because at time T you know the payout.

The PDEs all follow from the tower property. If $t_2 > t$ then (4), the tower property , and (5) (applied at time t_2) give

$$f(x,t) \;=\; E_{x,t}\left[f(X_{t_2},t_2)\right] \;. \tag{9}$$

Now suppose $t_2 = t + dt$ and expand the right side in a Taylor series as in Ito's lemma. The expectation is under the condition that $X_t = x$.

$$\begin{aligned}
E_{x,t}\left[f(X_{t+dt},t+dt)\right] &= E_{x,t}\left[f(x+dX+t,t+dt)\right] \\
&\approx E_{x,t}\left[f(x,t) + \partial_x f(x,t)dX_t + \tfrac{1}{2}\partial_x^2 f(x,t)dX_t^2 + \partial_t f(x,t)dt\right] \\
&= f(x,t) \,+\, \partial_x f(x,t)E_{x,t}[dX_t] \,+\, \tfrac{1}{2}\partial_x^2 f(x,t)E_{x,t}\left[dX_t^2\right] \,+\, \partial_t f(x,t)dt \\
&= f(x,t) \,+\, dt\Big\{\,\partial_t f(x,t) \,+\, a(x,t)\partial_x f(x,t) \,+\, \tfrac{b^2(x,t)}{2}\partial_x^2 f(x,t)\,\Big\}
\end{aligned}$$

Compare this with (9) and cancel $f(x,t)$ and then the dt from both sides and you get the *backward equation*

$$0 \;=\; \partial_t f(x,t) \,+\, a(x,t)\partial_x f(x,t) \,+\, \tfrac{b^2(x,t)}{2}\partial_x^2 f(x,t)\;. \tag{10}$$

The derivation of the backward equation has much in common with the derivation of Ito's lemma. We can make that more explicit by assuming that a function $f(x,t)$ satisfies the equation (10) and computing that the Ito differential $df(X_t,t) = 0$. This implies that if $t_2 > t$ (and even if it isn't actually) $f(X_{t_2},t_2) = f(X_t,t)$. Therefore $E[f(X_{t_2},t_2) \mid \mathcal{F}_t] = f(X_t,t)$. This allows us to conclude that if $f(x,t)$ satisfies the final condition (8) and the PDE (10) for all $t < T$, then f also satisfies (4)(5). All of the other backward equations below have derivations like the previous paragraph and verifications using Ito's lemma using this one. The advantage of the "derivation" is that it makes clear that the function f exists, at least for $t < T$.[1] This is significant, as we will see that a differentiable f that satisfies (8) probably does not exist for $t > T$.

The general theory of Markov processes has the concept of a *generator*, L. An operator operates on a function, $f(x)$, to produce another function, $g = Lf$. The generator of a diffusion is the *operator* appearing in (10), namely

$$L_t f(x) \;=\; a(x,t)\partial_x f(x) \,+\, \tfrac{b^2(x,t)}{2}\partial_x^2 f(x)\;.$$

In the common case where a and b do not depend on t, we drop the t subscript:

$$Lf(x) \;=\; a(x)\partial_x f(x) \,+\, \tfrac{b^2(x)}{2}\partial_x^2 f(x)\;. \tag{11}$$

[1] This is just the first mathematical step, however. The derivation does not prove that f is differentiable.

If f depends on t, the generator acts in the x variable, treating t as a parameter. The backward equation (10), in terms of the generator, is

$$\partial_t f \;=\; -Lf \; . \tag{12}$$

For any Markov process, the definition of the generator is

$$Lf \;=\; \lim_{t\to 0} \frac{1}{t}\left[f(X_t) - f(x) \right] \;,$$

assuming that $X_0 = X$. This is exactly the definition we implicitly used above.

The quantity (4) is not exactly right for finance because it does not discount the payout. A value function with discounting at a fixed rate from time T to time $t < T$ is

$$f(x,t) \;=\; e^{-r(T-t)} E_{x,t}\left[V(X_T) \right] \; . \tag{13}$$

There are several ways to derive a backward equation for this discounted expected payout. Here is one way, not the simplest. Apply the reasoning in the derivation of (10). The tower property, Taylor series, and some algebra following the earlier derivation gives

$$\begin{aligned} f(x,t) &= e^{-r\,dt}\left(e^{-r(T-(t+dt))} E_{x,t}\left[f(x+dX_t, t+dt\right]\right) \\ 0 &= -r\,dt f(x,t) \;+\; dt\left\{ \partial_t f(x,t) \;+\; a(x,t)\partial_x f(x,t) \;+\; \tfrac{b^2(x,t)}{2}\partial_x^2 f(x,t) \right\} . \end{aligned}$$

The resulting backward equation is

$$0 \;=\; \partial_t f(x,t) \;+\; a(x,t)\partial_x f(x,t) \;+\; \tfrac{b^2(x,t)}{2}\partial_x^2 f(x,t) \;-\; rf(x,t) \; . \tag{14}$$

A simpler derivation builds on our derivation of (10). Suppose g is the value function without discounting, $g(x,t) = E_{x,t}[V_X(T)]$. Then $f(x,t) = e^{-r(T-t)} g(x,t)$, with $\partial_t g = -Lg$. Simple differentiation with the product rule and the generator formula (12) gives

$$\begin{aligned} \partial_t f &= \partial_t\left(e^{-r(T-t)}\, g\right) \\ &= re^{-r(T-t)} g \;+\; e^{-r(T-t)}\partial_t g \\ &= rf \;-\; e^{-r(T-t)} Lg \\ \partial_t f &= rf \;-\; Lf \; . \end{aligned} \tag{15}$$

In getting the last line, we used the fact that L does not involve time derivatives, so $e^{-r(T-t)} Lg = L\left(e^{-r(T-t)} g\right) = Lf$. Of course, unwinding the definition of L turns this into (14). This trick, multiplication by exponential factors, will help us again.

Suppose instead you get a "running" payout over the interval $[0,T]$. This means that your total is $\int_0^T V(X_t)dt$. If we do not discount, it does not matter when the payments are actually made. The homework asks you to account for discounting. The appropriate value function is

$$f(x,t) \; = E_{x,t}\left[\int_t^T V(X_a) ds \right] \; .$$

Note that

$$\int_t^T V(X_s)ds \;=\; V(t)dt \;+\; \int_{t+dt}^T V(s)ds \; . \tag{16}$$

The tower property applied to this gives

$$f(x,t) \;=\; V(x,t)dt \;+\; E_{x,t}\left[\, f(x+dX_t, dt) \,\right] \; .$$

The backward equation derived from this is easily seen to be

$$0 \;=\; \partial_t f \;+\; Lf \;+\; V(x) \; . \tag{17}$$

This time the final condition is $f(x,T) = 0$.

Finally consider the quantity

$$f(x,t) \;=\; E_{x,t}\left[e^{\int_t^T V(X_s)ds} \right] \tag{18}$$

We will get models of this form when we use short rate models for the yield curve. The PDE for (18) is derived using the arguments for the previous two. The integral relation (16) gives

$$\begin{aligned} f(x,t) \;&=\; E_{x,t}\left[e^{\int_t^T V(X_s)ds} \right] \\ &=\; E_{x,t}\left[e^{V(x)dt} e^{\int_{t+dt}^T V(X_s)ds} \right] \\ &=\; (\,1 + V(x)dt\,)\, E_{x,t}\left[e^{\int_{t+dt}^T V(X_s)ds} \right] \\ &=\; (\,1 + V(x)dt\,)\, E_{x,t}[f(x+dX_t, t+dt)] \; . \end{aligned}$$

With this, the derivation of (15) leads to

$$\partial_t f(x,t) \;=\; Lf(x,t) \;+\; V(x)f(x,t) \; . \tag{19}$$

Clearly (look at (18)), this f also satisfies

$$f(x,T) \;=\; 1 \tag{20}$$

for all x.

These equations go by many names. All of them are backward equations. They sometimes are called *Kolmogorov* or *Chapman* (or both) backward equations. The expected value expression (18) is the *Feynmann-Kac* formula for the solution of (19) (20), because a special case of it was discovered by Marc Kac while he was trying to make sense of the *Feynman integral*. By now, the term is applied to many of these formulas and PDEs. You might, for example, hear (17) called "Feynman-Kac", even though Kac was interested in writing the solution of the PDE as an expectation, rather than finding the PDE satisfied by the expectation.

There are Ito's lemma verifications of the other PDEs. For instance, if you assume that f satisfies (19), you can calculate

$$df(X_t,t) \;=\; \left(\partial_t f(X_t,t) \,+\, Lf(X_t,t)\right)\,dt \;=\; -V(X_t)f(X_t,t)dt\;.$$

This integrates to[2]

$$f(X_T,T) \;=\; f(X_t,t)\,e^{-\int_t^T V(X_s,t)ds}\;.$$

Now you put the exponential term on the other side and take expectation in $\mathcal{F}_t$ and you get the solution formula (18).

2 The risk neutral process for the Black Scholes PDE

The following few paragraphs, as a matter of mathematics, would fit as comments at the end of the previous section. They earn their own section because of their importance in the theory of derivatives.

Recall from last week that the price of a European option satisfies the PDE

$$0 \;=\; \partial_t f(s,t) \,+\, \tfrac{1}{2}\sigma^2 s^2 \partial_s^2 f(s,t) \,+\, rs\partial_s f(s,t) \,-\, rf(s,t)\;. \tag{21}$$

This is the *Black Scholes equation.* It comes with final conditions

$$f(s,T) \;=\; V(s)\;, \tag{22}$$

where $V(s)$ is the payout. Now notice that the Black Scholes equation is the backward equation for the value function

$$f(s,t) \;=\; e^{-r(T-t)}E_{s,t}[V(S_T)]\;. \tag{23}$$

That is, if

$$dS_t \;=\; rS_t dt \,+\, \sigma S_t dW_t\;. \tag{24}$$

This is risk neutral pricing in the continuous time setting. Black and Scholes used the hedging argument from last week to show that the option price satisfies the PDE (21). Then they used the relationship between backward equations and diffusions to show that the solution to this PDE also can be represented as the discounted expected value (23). By definition, the process that enters into this expected value is the *risk neutral process.* The probability distribution on path

[2] It is a good sign if you worry about applying the rules of ordinary calculus here. But there is no Ito term because it is dt rather than dW or some other diffusion differential on the right. It is correct to write $df(X_t,t)/dt = V(X_t,t)f(X_t,t)$ and integrate.

space generated by the risk neutral process is the *risk neutral measure*, the *Q measure* for short.

Looking at the P-measure process we started with last week, the only difference in the risk neutral process is that the growth rate μ is replaced by the risk neutral rate r. As we said before when we did the binomial tree model, in a risk neutral measure, r is the growth rate of the stock, or any other traded asset. Indeed, the definition of the risk neutral world is that people are indifferent to risk. They value any uncertain risky asset at its discounted expected value. This includes also the underlying stock.

There are many situations where the PDE (21) or the stochastic representation are useful. For one thing, they give much information and intuition about even vanilla European prices. But for even vanilla American style options there are no closed form formulas. Any pricing methods use either the PDE and a time stepping method for it, or the stochastic representation and Monte Carlo, or some blatant and possibly risky approximation.

3 Dynamics and the direction of time

The term *dynamics* means a way to determine the state of a system at one time from the state at an earlier time. The most common ways to express dynamics are differential equations or difference equations. The dynamics for the backward equations of section 1, and particularly the Black Scholes equation (21) go backwards in time. That is, the PDE determines the value function $f(\cdot, t_1)$ in terms of the value function $f(\cdot, t_2)$, as long as $t_2 > t_1$. The expression $f(\cdot, t)$ refers to the function $f(s, t)$ as a function of s with the value t fixed. In more words, backward dynamics means that once $fs, t_2)$ is known for all values of s, then $f(s, t_1)$ can be determined for all values of s. It is not possible to determine $f(s, t_1$ for any value of s without knowing $f(s, t_2)$ for all values of s. You can think of $f(s, t_2)$ as the payout of the option at time t_2 as a function of the stock price. The Black Scholes says exactly this, that you can receive $f(s, t_2)$ for the option at time t_2 if $S_{t_2} = s$. That makes $f(s, t_2)$ exactly equivalent to a payout function.

You also could have this situation backwards: to have a value function $f(\cdot, t_1)$ and want a payout $V(\cdot)$ or a value function $f(\cdot, t_2)$ that produces $f(\cdot, t_1)$ (with $t_2 > t_1$). As we will see, it is rarely possible to do this. None of the backward equations likes to be run in forward time.

4 Finite difference time stepping

If you do not have a *closed form formula* for the solution of a PDE (which is most of the time), you can use a numerical method to compute approximate values of the solution. The two most important solution methods are Monte Carlo computation using the expectation representation such as (23)(24). But for low dimensional problems (Black Scholes is one dimensional), it almost always is better to use a deterministic method such as finite differences, finite elements, or the binomial tree. In my opinion, a finite difference time stepping method is usually better than the alternatives. You can find much discussion, much of it naive in my opinion, in Risk Magesine. As with most topics covered in this class, we discuss finite differences superficially. A serious production code for use in an industrial setting should use more sophisticated finite difference methods than the ones presented here.

A *finite difference* is a quotient that approximates a derivative. Examples we need are

$$\partial_t f(x,t) \approx \frac{f(x,t) - f(x,t-\delta t)}{\delta t} \tag{25}$$

$$\partial_t f(x,t) \approx \frac{f(x,t+\delta t) - f(x,t)}{\delta t} \tag{26}$$

$$\partial_x f(x,t) \approx \frac{f(x+\delta x,t) - f(x,t)}{\delta x} \tag{27}$$

$$\partial_x f(x,t) \approx \frac{f(x,t) - f(x-\delta x,t)}{\delta x} \tag{28}$$

$$\partial_x f(x,t) \approx \frac{f(x+\delta x,t) - f(x-\delta x,t)}{2\delta x} \tag{29}$$

$$\partial_x^2 f(x,t) \approx \frac{f(x+\delta x,t) - 2f(x,t) + f(x-\delta x,t)}{\delta x^2} . \tag{30}$$

You can see that there is more than one way to estimate, say, $\partial_t f$. We choose the approximation formula that helps us achieve a given task. The first two approximations are *first order accurate*, which means that the error (the difference between the right and left sides) is roughly proportional to the step size (δt for the first two and δx for the next two). The last two approximations are second order accrurate. This usually means much more accurate.

A finite difference marching method for a backward equation uses a finite difference consisting of *grid points* (x_j, t_k), with $x_j = j\delta x$ and $t_k = k\delta t$. As usual, we suppose that the final time is $T = t_n = n\delta t$. The finite difference method computes approximations $f_{j,k} \approx f(x_j, t_k)$. Given the values $f_{j,k}$, one *time step* of the method computes the values $f_{j,k-1}$. This is done using well chosen finite difference approximations to all the derivatives in the PDE, say, (10). Focus on one grid point (x_j, t_k). Approximate (10) at that point using the *backward* time difference (25). Do this using $t_k - \delta t = t_{k-1}$ and the approximate values, and you get

$$\partial_t f(x_j, t_k) \approx \frac{f(x_j, t_k) - f(x_j, t_{k-1})}{\delta t} \approx \frac{f_{j,k} - f_{j,k-1}}{\delta t}$$

In the same way, we can use (29) to get

$$\partial_x f(x_j, t_k) \approx \frac{f_{j+1,k} - f_{j-1,k}}{2\delta x} ,$$

and (30) to get

$$\partial_x^2 f(x_j, t_k) \approx \frac{f_{j+1,k} - 2f_{j,k} + f_{j+1,k}}{\delta x^2} .$$

Until now we still have not defined the approximate values $f_{j,k}$. The time stepping method defines then using the finite difference approximation to the PDE using the above finite difference approximations to the derivatives:

$$0 = \frac{f_{j,k} - f_{j,k-1}}{\delta t} + a(x_j, t_k)\frac{f_{j+1,k} - f_{j-1,k}}{2\delta x} + \tfrac{1}{2}b^2(x_j, t_k)\frac{f_{j+1,k} - 2f_{j,k} + f_{j+1,k}}{\delta x^2}$$

Now, the purpose of all this was to compute the numbers $f_{j,k-1}$ from the numbers $f_{j,k}$. For this, you only need to put $f_{j,k-1}$ on the other side of the equation and solve:

$$f_{j,k-1} = f_{j,k} + \frac{\delta t}{2\delta x}a(x_j, t_k)\,(\,f_{j+1,k} - f_{j-1,k}\,) + \frac{\delta t}{2\delta x^2}b^2(x_j, t_k)\,(\,f_{j+1,k} - 2f_{j,k} + f_{j+1,k}\,) \tag{31}$$

This is the *forward Euler* method, also called *explicit* Euler, for finding approximate solutions of (10). The other backward equations are very similar. For example, we solve (19) simply by adding $\delta t V(x_j) f_{j,k}$ to the left side of (31). You use the final conditions to start the process. For (10), this means taking

$$f_{j,n} = V(x_j) \tag{32}$$

for all j. The equations (31) then compute the numbers $f_{j,n-1}$, then the numbers $f_{j,n-2}$, and so on until $n = 0$.

This method is called *explicit* because (31) is an explicit formula for $f_{j,k-1}$ in terms of already determined numbers $f_{j,k}$. An alternative would be an *implicit* method in which the time stepping finite difference approximations do not determine the new numbers explicitly in terms of the old ones. Instead one has to solve a system of equations for the new numbers at each time step. Clearly, there would have to be a good reason to do this, and there is. The implicit method is more stable (see below). The simplest implicit method is the *backward Euler* or *implicit Euler* method, which comes from using (26) instead of (25). In this case, we would have $f_{j,k+1}$ on the right of (31). That would be known and the left side would form a system of simultaneous equations for the so far unknown $f_{j,k}$.

It is easy to get confused with the terms forward and backward above. Both the forward and backward Euler methods move backwards in time, computing $f_{\cdot,k-1}$ from $f_{\cdot,k}$. The terms forward and backward come from using the forward or backward time derivatives (25) or (26). The problem with this is that the forward Euler method uses what usually would be called the backward difference formula (25), and the backward Euler method uses the forward time difference formula (26). This confusion comes from the fact that these methods were developed for physical problems in which you march time into the future, not into the past. In that case, the explicit method uses the forward difference and the implicit method uses the backward difference. The association forward Euler = explicit Euler is too hard to change, even for an industry as important as finance.

If you apply the method (31) with what you know now, the computation is likely to *blow up* rather than produce anything useful. That is because, if you are not careful, the method will be *unstable*. Instability means that errors at time k will be amplified by a factor larger than one as they are passed to time $k-1$. If there is a growth factor of 2, for instance, an error introduced at the final time is amplified by a factor of 2^n by the time the computation is over. Taking $n = 40$ time steps makes this amplification factor $2^{40} = \left(2^{10}\right)^4 \approx \left(10^3\right)^4 = 10^{12}$. In order to give good approximations, a finite difference method must use accurate finite difference equations, and be stable.

This class does not have time for a detailed discussion of stability so we just give conditions that ensure that our forward Euler approximation (31) is stable. To state the condition, note that with some algebra you can write (31) in the general form

$$f_{j,k-1} \;=\; \alpha_{j,k} f_{j-1,k} \,+\, \beta_{j,k} f_{j,k} \,+\, \gamma_{j,k} f_{j+1,k} \;. \tag{33}$$

The formulas for the coefficients are

$$\alpha_{j,k} \;=\; -\frac{\delta t}{2\delta x} + \frac{\delta t}{2\delta x^2} b^2(x_j, t_k) \tag{34}$$

$$\beta_{j,k} \;=\; 1 - \frac{\delta t}{\delta x^2} b^2(x_j, t_k) \tag{35}$$

$$\gamma_{j,k} \;=\; \frac{\delta t}{2\delta x} + \frac{\delta t}{2\delta x^2} b^2(x_j, t_k) \;. \tag{36}$$

The stability condition is that coefficients all are positive

$$\alpha_{j,k} \;\geq 0 \;\;, \quad \beta_{j,k} \;\geq 0 \;\;, \quad \gamma_{j,k} \;\geq 0 \;. \tag{37}$$

It is a simple algebraic calculation that

$$\alpha_{j,k} \,+\, \beta_{j,k} \,+\, \gamma_{j,k} \;=\; 1 \;. \tag{38}$$

This, plus the positivity conditions (37), imply that

$$|f_{j,k-1}| \;\leq\; \max\{\, |f_{j-1,k}| \,,\, |f_{j,k}| \,,\, |f_{j+1,k}| \,\}$$

Clearly, this implies that

$$\max_j |f_{j,k-1}| \;\leq\; \max_j |f_{j,k}| \;. \tag{39}$$

Arguing by induction on k, this means that the solution values $f_{j,k}$ are within the range of the final values $V(x_j)$. This is a strong sense in which the difference equations (31) are stable.

Once the *space step*, δx, is chosen, the stability constraints (37) place a serious limit on the time step. From (35), we see that

$$\delta t \;\leq\; b^2(x_j, t_k)\, \delta x^2 \;. \tag{40}$$

This usually means that if you reduce δx by a factor of 2, then you have to reduce δt by a factor of 4. In other words, is you make the space step small, you have to make the *time step*, δt, very small. This is bad because the total work is proportional to the number of time steps, which is inversely proportional to δx.

To be clear, I review the above argument backwards. A small step size requires a very small time step, and an expensive computation, because otherwise the method would be unstable.[3] The method is unstable when one of the coefficients (34), (35), or (36) is negative. The first of these conditions to be violated, particularly in the normal case $b^2 > 0$, usually is the β condition. That is because if $\delta t \leq O(\delta x^2)$, then the ratio $\delta t a(x_j, t_k)/2\delta x$ is much smaller than $\delta t b(x_j, t_k)/2\delta x^2$.

The time step constraint (40) does have one happy side effect. It makes the overall method second order accurate in terms of δx. We used second order accurate for both space derivatives, but the error in the difference formula for $\partial_t f$ was of order δt. Now that δt is of order δx^2, this gives the overall method an overall error of order δx^2. This will be discussed in much more detail in Scientific Computing and Computational Methods in Finance.

The formulas (33) resemble the binomial tree formulas, except that here we use the middle value as well as the outer values. For this reason, the forward Euler method often is called the *trinomial tree* method. This makes sense particularly if the coefficients (34), (35), and (36) are positive. In that case, the finite difference equations (33) are the discrete backward equations for a trinomial tree process. This process has random walk $X_k^{\delta t}$ with rules

$$X_{k-1}^{\delta t} \rightarrow \begin{cases} X_k^{\delta t} = X_{k-1}^{\delta t} + \delta t & \text{with probility } \gamma_{j,k} \\ X_k^{\delta t} = X_{k-1}^{\delta t} & \text{with probility } \beta_{j,k} \\ X_k^{\delta t} = X_{k-1}^{\delta t} + \delta t & \text{with probility } \alpha_{j,k} \end{cases} \tag{41}$$

The probability rules (reftt) define the *trinomial tree process*. Note that the largest possible time step is the one that makes the time step inequality (40) an equality. This makes $\beta_{j,k} = 0$ and turns the trinomial tree into a binomial tree. In other words, taking the largest time step possible turns the trinomial tree into the binomial tree. You might thing we always would do this in the interest of taking the smallest number of time steps, but there are many reasons not to do this, as we will see.

A final "detail" you might have been wondering about concerns the range of the x variable index j. Clearly j cannot run from $-\infty$ to ∞ on a computer. The resolution of this problem will occupy some time in the Computational Methods in Finance class. But here we offer a cheap/expensive solution – cheap because it takes little thought, expensive because it uses far too much computer time. Shrink the mesh in space by one grid point on either side for each time step.

[3] I omit the proof that violating (40) makes the method unstable. If a and b are constants, you can see this using the Fourier transform and in other less direct ways.

5 The log variable transformation

The Black Scholes equation is easier to solve if you use the log variable transformation $x = \log s$. Doing the chain rule computation gives

$$0 \;=\; \partial_t f \;+\; \frac{\sigma^2}{2}\partial_x^2 f \;+\; \left(r - \frac{\sigma^2}{2} \right) \partial_x f \;-\; rf \;. \tag{42}$$

Now, a uniform grid in the x variable as discussed above turns into a geometric mesh in the s variable. The forward Euler method with the largest possible time step is exactly the binomial tree method discussed earlier.

6 American style options and early exercise

Most equity options are American style with an early exercise feature. The value of such an option cannot fall below it's *intrinsic*, or *early exercise*, value because that would be an arbitrage. On the other hand, whenever the price of an American option is above the intrinsic value, that price must satisfy the Black Scholes PDE because the arbitrage argument applies. This, pricing American style options leads to solving a sort of PDE with an inequality constraint. European style options, on the other hand, may have prices below the intrinsic value. It is quite possible that the holder of a European options may wish he/she could exercise early.

The way to price Americans (that is, American style options) in the forward Euler context is simply to take the max at each time step with the intrinsic value. Call the intrinsic value $V(s)$ (usually the early exercise payout and the final time payout are the same). Then there are some points where $f_{j,k} = V(x_j)$ and others where $f_{j,k} > V(x_j)$. The former are in the *early exercise region* and the latter are in the *hold region*, also called the *continuation region*. Their mutual boundary is the *early exercise boundary*. The optimal strategy if you own an American style is to exercise at the first time the underlier touches the early exercise boundary.

In the finite difference method just described, the early exercise region at any given time is the set of grid points where the finite difference approximation thinks you should exercise early. In the limit as $\delta x \to 0$, this numerical early exercise region should converge to the exact region. As with the value function, the smaller δx becomes, the more accurate the computed early exercise boundary.

An interesting thing about American style options is the *smooth pasting* condition at the early exercise boundary. This says that the Δ of the value function on the left and right side of the early exercise point agree. For example, suppose $S_*(t)$ is the early exercise price (also called *critical price*) for an American style put. The early exercise region is $S_t \leq S_*(t)$. The value function is $f(s,t) = V(s) = (K-s)_+$ for $s < S_*(t)$. Clearly $\Delta = -1$ there. Smooth pasting states that $\Delta(s,t) = \partial_s(f(s,t) \to -1$ as $s \to S_*(t)$ from above. On the other hand, $\Gamma = \partial_s^2 f(s,t)$ is not continuous at $S_*(t)$.

7 Numerical Greeks

We had formulas for the Greeks of vanilla European puts and calls. For American style options, we can estimate the Greeks also by taking finite difference approximations to the derivatives using the approximate value function. For example, suppose $x = \log(s)$ and $g(x,t)$ is the value function in the log variable, so that the value function in the s variable is $f(s,t) = g(\log(s),t)$. We can calculate

$$\Gamma(s,t) \;=\; \partial_s^2 g(\log(s),t) \;=\; \frac{1}{s^2}\partial_x^2 g(\log(s),t) \;-\; \frac{1}{s^2}\partial_x g(\log(s),t)\;.$$

The derivatives of g can be estimated as

$$\partial_x g(x_j,t_k) \;\approx\; \frac{g_{j+1,k} - g_{j-1,k}}{2\delta x}\;,$$

and similarly for $\partial_x^2 g(x,t)$.

It is harder to find the derivatives $\Theta = \partial_\sigma f$ and $\rho = \partial_r f$. One way is to write the PDE that the Greek satisfies. For example we can differentiate the log variable equation (42) with respect to σ to get

$$0 \;=\; \partial_t\Lambda \;+\; \frac{\sigma^2}{2}\partial_x^2\Lambda \;+\; \left(r - \frac{\sigma^2}{2}\right)\partial_x\Lambda \;-\; r\Lambda \;+\; \sigma\partial_x^2 f - \sigma\partial_x f\;. \qquad (43)$$

The last two terms could be found from a previous finite difference computation of f itself.

This is one of the payoffs of using a finite difference method rather than a binomial tree. In the finite difference method, the mesh need not depend on the parameters. This means that the solution can depend in a smooth way on parameters. In the binomial tree method, the tree itself depends on σ.