PDE for Finance Notes – Section 7

Notes by Robert V. Kohn, Courant Institute of Mathematical Sciences. For use only in connection with the NYU course PDE for Finance, MATH-GA 2706. The typed part of Section 7 dates from 2003, but the discussion of the Garleanu-Pedersen paper (in handwritten notes) is new in 2015.

Discrete-time dynamic programming. This section achieves two goals at once. One is to demonstrate the utility of discrete-time dynamic programming as a flexible tool for decision-making in the presence of uncertainty. The second is to introduce some financially-relevant applications. To achieve these goals Section 7 presents three examples: (1) optimal control of execution costs (following a paper by D. Bertsimas and A. Lo, *J Financial Markets* 1, 1998, 1-50); (2) least-square replication of a European option (following a paper by D. Bertsimas, L. Kogan, and A. Lo, *Operations Research* 49 (2001) 372-397; and (3) optimization of risk vs return in a multiperiod model with transaction costs (following a paper by N. Garleanu and L.H. Pedersen, *J Finance* 68, 2013, 2309-2340. (Examples 1 and 2 are typed text; the discussion of example 3 is in separate handwritten notes.)

First, some remarks by way of orientation. In the context of this course it was natural to address continuous-time problems first, because we began the semester with stochastic differential equations and their relation to PDE's. Most courses on optimal control would however discuss the discrete-time setting first, because it is in many ways easier and more flexible. Indeed, continuous-time dynamic programming uses stochastic differential equations, Ito's formula, and the HJB equation. Discrete-time dynamic programming on the other hand uses little more than basic probability and the viewpoint of dynamic programming. Of course many problems have both discrete and continuous-time versions, and it is often enlightening to consider both (or compare the two). A general discussion of the discrete-time setting, with many examples, can be found in Dimitri Bertsekas, *Dynamic Programming: Deterministic and Stochastic Models*, Prentice-Hall, 1987, especially Chapter 2. Our approach here is different: we shall explain the method by presenting a few financially-relevant examples.

Example 1: Optimal control of execution costs. This example is taken from the recent article: Dimitris Bertsimas and Andrew Lo, *Optimal control of execution costs*, *J. Financial Markets* 1 (1998) 1-50 (available through the NYU library system, also easy to find on Bertsimas' website using Google Scholar).

The problem is this: an investor wants to buy a large amount of some specific stock. If he buys it all at once he'll drive the price up, thereby paying much more than necessary. Better to buy part of the stock today, part tomorrow, part the day after tomorrow, etc. until the full amount is in hand. But how best to break it up?

Here's a primitive model. It's easy to criticize (we'll do this below), but it's a good starting point – and an especially transparent example of stochastic optimal control. Suppose the investor wants to buy $S_{\rm tot}$ shares of stock over a period of N days. His control variable is S_i, the number of shares bought on day i. Obviously we require $S_1 + \ldots + S_N = S_{\rm tot}$.

We need a model for the impact of the investor's purchases on the market. Here's where this model is truly primitive: we suppose that the price P_i the investor achieves on day i is related to the price P_{i-1} on day $i-1$ by

$$P_i = P_{i-1} + \theta S_i + \sigma e_i \tag{1}$$

where e_i is a Gaussian random variable with mean 0 and variance 1 (independent of S_i and P_{i-1}). Here θ and σ are fixed constants.

And we need a goal. Following Bertsimas and Lo we focus on minimizing the expected total cost:

$$\min E\left[\sum_{i=1}^{N} P_i S_i\right].$$

To set this up as a dynamic programming problem, we must identify the *state*. There is a bit of art here: the principle of dynamic programming requires that we be prepared to start the optimization at any day $i = N, N-1, N-2, \ldots$ and when $i = 1$ we get the problem at hand. Not so hard here: the state on day i is described by the most recent price P_{i-1} and the amount of stock yet to be purchased $W_i = S_{\text{tot}} - S_1 - \ldots - S_{i-1}$. The state equation is easy: P_i evolves as specified above, and W_i evolves by

$$W_{i+1} = W_i - S_i.$$

Dynamic programming finds the optimal control by starting at day N, and working backward one day at a time. The relation that permits us to work backward is the one-time-step version of the principle of dynamic programming. In this case it says:

$$V_i(P_{i-1}, W_i) = \min_s E\left[P_i s + V_{i+1}(P_i, W_{i+1})\right].$$

Here $V_i(P, W)$ is the value function:

$$\begin{aligned} V_i(P, W) \quad = \quad & \text{optimal expected cost of purchasing } W \text{ shares} \\ & \text{starting on day } i, \text{ if the most recent price was } P. \end{aligned}$$

(The subscript i plays the role of time.)

To find the solution, we begin by finding $V_N(P, W)$. Since $i = N$ the investor has no choice but to buy the entire lot of W shares, and his price is $P_N = P + \theta W + e_N$, so his expected cost is

$$V_N(P, W) = E\left[(P + \theta W + \sigma e_N)W\right] = PW + \theta W^2.$$

Next let's find $V_{N-1}(P, W)$. The dynamic programming principle gives

$$\begin{aligned} V_{N-1}(P, W) &= \min_s E\left[(P + \theta s + \sigma e_{N-1})s + V_N(P + \theta s + \sigma e_{N-1}, W - s)\right] \\ &= \min_s E\left[(P + \theta s + \sigma e_{N-1})s + (P + \theta s + \sigma e_{N-1})(W - s) + \theta(W - s)^2\right] \\ &= \min_s \left[(P + \theta s)s + (P + \theta s)(W - s) + \theta(W - s)^2\right] \\ &= \min_s \left[W(P + \theta s) + \theta(W - s)^2\right]. \end{aligned}$$

The optimal s is $W/2$, giving value

$$V_{N-1}(P,W) = PW + \frac{3}{4}\theta W^2.$$

Thus: starting at day $N-1$ (so there are only 2 trading days) the investor should split his purchase in two equal parts, buying half the first day and half the second day. His impact on the market costs him, on average, an extra $\frac{3}{4}\theta W^2$ over the no-market-impact value PW.

Proceeding similarly for day $N-2$ etc., a pattern quickly becomes clear: starting at day $N-i$ with the goal of purchasing W shares, if the most recent price was P, the optimal trade on day i (the optimal s) is $W/(i+1)$, and the expected cost of all W shares is

$$V_{N-i}(P,W) = WP + \frac{i+2}{2(i+1)}\theta W^2.$$

This can be proved by induction. The inductive step is very similar to our calculation of V_{N-1}, and is left to the reader.

Notice the net effect of this calculation is extremely simple: no matter when he starts, the investor should divide his total goal W into equal parts – as many as there are trading days – and purchase one part each day. Taking $i = N-1$ we get the answer to our original question: if the most recent price is P and the goal is to buy S_{tot} over N days, then this optimal strategy leads to an expected total cost

$$V_1(P,S_{\text{tot}}) = PS_{\text{tot}} + \frac{\theta}{2}(1+\frac{1}{N})S_{\text{tot}}^2.$$

There's something unusual about this conclusion. The investor's optimal strategy is not influenced by the random fluctuations of the prices. It's always the same, and can be fixed in advance. That's extremely unusual in stochastic control problems: the optimal control can usually be chosen as a *feedback control*, i.e. a deterministic function of the state – but since the state depends on the fluctuations, so does the control.

I warned you it was easy to criticize this model. Some comments:

1. The variance of the noise in the price model never entered our analysis. That's because our hypothetical investor is completely insensitive to risk – he cares only about the expected result, not about its variance. No real investor is like this.

2. The price law (1) is certainly wrong: it has the ith trade S_i increasing not just the ith price P_i but also *every subsequent price*. A better law would surely make the impact of trading *temporary*. Bertismas and Lo consider one such law, for which the problem still has a closed-form solution derived by methods similar to those used above.

The take-home message: Discrete-time stochastic dynamic programming is easy and fun. Of course a closed-form solution isn't always available. When there is no closed-form solution one must work backward in time *numerically*. The hardest part of the whole thing is keeping your indices straight, and remembering which information is known at time i, and which is random.

****

Example 2: Least-square replication of a European option. This discussion follows the paper D. Bertsimas, L. Kogan, and A.W. Lo: *Hedging derivative securities and incomplete markets: an ϵ-arbitrage approach*, Operations Research 49 (2001) 372-397. (Available through the NYU library system, via JSTOR; also easy to locate on Andrew Lo's website using Google Scholar). The paper is quite rich; I focus for specificity on the simplest case, when the returns at different times are independent trials from a single probability distribution. However you'll see as we go along that this hypothesis isn't really being used; the method is actually much more general. (I'll comment on its scope at the end.)

Here's the problem. Consider a stock that can be traded only at discrete times $j\Delta t$, and suppose its price P_j at the jth time satisfies

$$P_j = P_{j-1}(1 + \phi_{j-1}) \tag{2}$$

where ϕ_{j-1} is chosen from a specified distribution, independent of j. (The discrete-time analogue of standard lognormal dynamics is obtained by taking $\log(1+\phi_j) = \mu\Delta t + \sigma\sqrt{\Delta t}e$ where e is Gaussian with mean 0 and variance 1.) You are an investment banker, and at time $j = 0$ you sell an option with maturity N and payout $F(P_N)$, receiving cash V_0 in payment. Your goal is to invest this cash wisely, trading in a self-financing way, so the value at the final time comes as close as possible to replicating the payout $F(P_N)$.

The *state* at time j is

$$\begin{aligned} V_j &= \text{the value of your portfolio at time } j, \text{ and} \\ P_j &= \text{the price at which trades can be made at time } j. \end{aligned}$$

We suppose that on each day, knowing V_j and P_j (but not the next day's price P_{j+1}) you make a decision how to rebalance your portfolio, buying or selling at price P_j till you hold θ_j units of stock and B_j units of cash. Thus θ_j is the *control*. Each trade must be "self-financing." To understand what this means, observe that going into the jth day your portfolio is worth

$$\theta_{j-1}P_j + B_{j-1}$$

while after rebalancing it is worth

$$\theta_j P_j + B_j.$$

For the trade to be self-financing these two expressions must be equal; this gives the restriction

$$P_j(\theta_j - \theta_{j-1}) + (B_j - B_{j-1}) = 0.$$

Since the value of your portfolio on the jth day is

$$V_j = \theta_j P_j + B_j,$$

the value changes from day to day by the law

$$V_j - V_{j-1} = \theta_{j-1}(P_j - P_{j-1}).$$

We interpret the goal of replicating the payout "as well as possible" in a least-square sense: your aim is to choose the θ_j's so as to

$$\text{minimize } E\left[(V_N - F(P_N))^2\right].$$

This time it is fairly obvious how to fit the problem into the dynamic programming framework. At time i the value function is

$$J_i(V,P) = \min_{\theta_i,\ldots,\theta_{N-1}} E_{V_i=V,\,P_i=P}\left[|V_N - F(P_N)|^2\right].$$

The final-time condition is

$$J_N(V,P) = |V - F(P)|^2$$

since on day N there is no decision to be made. The principle of dynamic programming gives

$$J_i(V,P) = \min_{\theta_i} E_{P_i=P}\left[J_{i+1}(V + \theta_i(P_{i+1} - P_i), P_{i+1})\right].$$

Now a small miracle happens (this is the advantage of the least-square formulation): *the value function is, at each time, a quadratic polynomial in V, with coefficients which are computable functions of P.* In fact:

Claim: The value functions have the form

$$J_i(V_i,P_i) = a_i(P_i)|V_i - b_i(P_i)|^2 + c_i(P_i)$$

and the optimal control θ_i is given by a feedback law that's linear in V_i:

$$\theta_i(V_i,P_i) = p_i(P_i) - V_i q_i(P_i).$$

The functions p_i, q_i, a_i, b_i, and c_i are determined inductively by the following explicit formulas:

$$\begin{aligned}
p_i(P_i) &= \frac{E\left[a_{i+1}(P_{i+1})\cdot b_{i+1}(P_{i+1})\cdot(P_{i+1}-P_i)\right]}{E\left[a_{i+1}(P_{i+1})\cdot(P_{i+1}-P_i)^2\right]}\\
q_i(P_i) &= \frac{E\left[a_{i+1}(P_{i+1})\cdot(P_{i+1}-P_i)\right]}{E\left[a_{i+1}(P_{i+1})\cdot(P_{i+1}-P_i)^2\right]}\\
a_i(P_i) &= E\left[a_{i+1}(P_{i+1})\cdot[1-q_i(P_i)(P_{i+1}-P_i)]^2\right]\\
b_i(P_i) &= \frac{1}{a_i(P_i)}E\left[a_{i+1}(P_{i+1})\cdot[b_{i+1}(P_{i+1}) - p_i(P_i)(P_{i+1}-P_i)]\cdot[1-q_i(P_i)(P_{i+1}-P_i)]\right]\\
c_i(P_i) &= E\left[c_{i+1}(P_{i+1})\right] - a_i(P_i)\cdot b_i(P_i)^2 + E\left[a_{i+1}(P_{i+1})\cdot[b_{i+1}(P_{i+1}) - p_i(P_i)(P_{i+1}-P_i)]^2\right]
\end{aligned}$$

where all expectations are over the uncertainties associated with passage from time i to $i+1$. These relations can be solved backward in time, starting from time N, using the initialization

$$a_N(P_N) = 1, \quad b_N(P_N) = F(P_N), \quad c_N(P_N) = 0.$$

Play before work. Let's explore the impact of the claim.

Main consequence: The price you charged for the option – V_0 – never enters the analysis. But of course it's not irrelevant! If you charged V_0 for the option and the day-0 price was P_0, then your expected replication error is $J_0(V_0, P_0) = a_0(P_0)|V_0 - b_0(P_0)|^2 + c_0(P_0)$. The first term is always positive, so *the price that minimizes the replication error is* $V_0 = b_0(P_0)$.

Is $V_0 = b_0(P_0)$ necessarily the market price of the option? Not so fast! This would be so – by the usual absence-of-arbitrage argument – if $c_0(P_0)$ were zero, since in that case the payout is exactly replicatable. However in general $c_0(P_0)$ is positive. (It is clearly nonnegative, since the mean-square replication error is nonnegative no matter what the value of V_0. It is generally positive, due to market incompleteness: even the Black-Scholes marketplace is not complete in the discrete-time setting.) If c_0 is small then the price of the option should surely be close to b_0. However there is no logical reason why it should be exactly b_0. For example, the *sign* of the replication error $V_N - F(P_N)$ makes a great deal of difference to the seller (and to buyer) of the option, but it did not enter our discussion. Moreover there is no reason a specific investor should accept the *quadratic* replication error as the appropriate measure of risk.

What about the Black-Scholes marketplace, where the classical Black-Scholes-Merton analysis tells us how to price and hedge an option? That analysis is correct, of course, but it requires trading *continuously* in time. If you can trade only at discrete times $j\Delta t$ then the market is no longer complete and options are not exactly replicatable. If you use the optimal trading strategy determined in our Claim, your mean-square replication error will be *smaller than* the value obtained by using the continuous-time Black-Scholes hedging strategy (which sets $\theta_j = \partial V/\partial P$ evaluated at $P = P_j$ and $t = j\Delta t$, where V solves the Black-Scholes PDE). How much smaller? This isn't quite clear, at least not to me. The paper by Bertsimas-Kogan-Lo does show, however, that the discrete-time results converge to those of the continuous-time analysis as $\Delta t \to 0$.

OK, now work. We must justify the claim. Rather than do a formal induction, let us simply explain the first step: why the formulas are correct when $i = N - 1$. This has all the ideas of the general case, and the notation is slightly simpler since in this case $a_{i+1} = a_N = 1$. The principle of dynamic programming gives

$$J_{N-1}(V_{N-1}, P_{N-1}) = \min_{\theta_{N-1}} E\left[|V_{N-1} + \theta_{N-1}(P_N - P_{N-1}) - F(P_N)|^2\right].$$

Simplifying the notation, let us write the right hand side as

$$\min_{\theta} E\left[|V + \theta\delta P - F|^2\right], \tag{3}$$

bearing in mind that $\delta P = P_N - P_{N-1}$ and $F = F(P_N)$ are random variables, and V and θ are deterministic constants.

Identification of the optimal θ is essentially a task of linear algebra, since

$$\langle \xi, \eta \rangle = E[\xi\eta]$$

is an inner product on the vector space of random variables. We need to view the constant function V as a random variable; let us do this by writing it as $V\mathbf{1}$ where V is scalar and $\mathbf{1}$ is the random variable which always takes the value 1. Then (3) can be written as

$$\min_\theta \|V\mathbf{1} + \theta\delta P - F\|^2$$

where $\|\xi\|^2 = \langle\xi,\xi\rangle = E[\xi^2]$. Decomposing $\mathbf{1}$ and F into the parts parallel and orthogonal to δP, we have

$$\mathbf{1} = (\mathbf{1} - q\delta P) + q\delta P \quad \text{with } q = \langle\mathbf{1}, \delta P\rangle\|\delta P\|^{-2}$$

and

$$F = (F - p\delta P) + p\delta P \quad \text{with } p = \langle F, \delta P\rangle\|\delta P\|^{-2},$$

and

$$\begin{aligned}\|V\mathbf{1} + \theta\delta P - F\|^2 &= \|V(\mathbf{1} - q\delta P) - (F - p\delta P) + (\theta + Vq - p)\delta P\|^2 \\ &= \|V(\mathbf{1} - q\delta P) - (F - p\delta P)\|^2 + (\theta + Vq - p)^2\|\delta P\|^2.\end{aligned}$$

The optimal θ makes the second term vanish: $\theta = p - Vq$, and the resulting value is

$$V^2\|\mathbf{1} - q\delta P\|^2 - 2V\langle\mathbf{1} - q\delta P, F - p\delta P\rangle + \|F - p\delta P\|^2$$

This is, as expected, a quadratic polynomial in V, which can be written in the form $a(V - b)^2 + c$. Expressing p, q, a, b and c in the original probabilistic notation gives precisely the formulas of the Claim with $i = N - 1$. The general inductive step is entirely similar.

Let's close by discussing the scope of this method. The case considered above – returns that are independent and identically distributed at each time step – is already of real interest. It includes the time discretization of the standard Black-Scholes marketplace, but it is much more general. For example, it can also be used to model a stock whose price process has jumps (see Section 3.3 of Bertsimas-Kogan-Lo).

Moreover the framework is by no means restricted to the case of such simple price dynamics. All one really needs is that (i) the price is determined by a Markov process, and (ii) the payout at maturity depends on the final state of this process. Thus the same framework can be used for problems as diverse as:

- An exotic option whose payout is the maximum stock price between times 0 and N. Just replace the stock price process P_j with the process (P_j, M_j) = (price at time j, max price through time j), defined by

  $$P_j = P_{j-1}(1 + \phi_{j-1}), \quad M_j = \max\{P_j, M_{j-1}\}$$

  and replace the payout $F(P_N)$ by one of the form $F(M_N)$.

- Stochastic volatility. Just replace the stock price process P_j with a process (P_j, σ_j) = (price at time j, volatility at time j), with the time-discretization of your favorite stochastic-volatility model as the dynamics. The payout would, in this case, still have the form $F(P_N)$.

****

Example 3: Optimization of risk vs return in a multiperiod model with transaction costs. This example comes from a recent paper by N. Garleanu and L.H. Pedersen, *J Finance* 68, 2013, 2309-2340 (available through the NYU Library system; also easy to find on Lasse Pedersen's website http://www.lhpedersen.com/home). The paper considers a universe with finitely many stocks (so the issue is portfolio optimization). I'll discuss just the special case where there is a single stock (so the issue is how much stock to hold at each time step). See the accompanying handwritten notes (posted as Section 7, part 2).