Mathematical & Computational Finance II
Lecture Notes

PDE Methods for Option Pricing

November 17 2015
Last update: December 4, 2017

1 American Options

1.1 Preliminaries

Suppose the payoff of an option written on asset with price process S is $\Gamma(S)$ and the option can be exercised at any time $t \in [0, T]$. Then, the value of the American option, denoted by

$$V_{AM}(S, t) \quad \text{at time } t \in [0, T]$$

must satisfy

$$V_{AM}(S, t) \geq \Gamma(S)$$

On the other hand, note that we may always wait until the final time T to exercise the option. Therefore, we must have the value

$$V_{AM}(S, t) \geq V_{Euro}(S, t)$$

Lets consider the case of a put option. At some point the value of the European put may dip below the payoff (i.e. if we expect that $\mathbb{P}(\textit{Go back in the money})$ to be small). This cannot happen for an American put since you can just buy it and immediately exercise it for a riskless profit $(S - S_t) - V^P_{AM}(S, t)$ at time t. So, in the case of an American put option, the value most satisfy

$$\begin{aligned} V^P_{AM}(S, t) &> K - S \quad \text{if } S > S_f(t) \\ V^P_{AM}(S, t) &= K - S \quad \text{if } S \leq S_f(t) \end{aligned}$$

for some "contact point" $S_f(t)$, where the boundary $S_f(t)$ is some smooth curve below K (in the case of a put) which dictates at what price S can drop to when the put value is identically $K - S$. We can show that $V_{AM}(S, t)$ must satisfy something called the smooth pasting condition at the boundary:

$$\frac{\partial}{\partial S} V_{AM}\left(S_f(t), t\right) = \frac{d\Gamma}{dS}\left(S_f(t)\right)$$

Take a portfolio Π of

$$\Pi = \begin{cases} \text{Long 1 option} \\ \text{Short } \Delta \text{ units of } S \end{cases}$$

Then the value of this portfolio, denoted Π_t, is

$$\Pi_t = V_t - \Delta S_t$$

Hence, for an American option, the holder can make a risk free profit in excess of r_f unless we satisfy[1]

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV \leq 0$$

"This is what we call a linear complimentarity problem."

Thus, we must have that either

$$\begin{aligned} &V_{AM} = \Gamma(S) \quad \text{or} \\ &V_{AM} \text{ satisfies the Black-Scholes inequality above}^2 \end{aligned}$$

No arbitrage implies that the Black-Scholes inequality must be satisfied. The optimal solution satisfies all above inequalities and either $V_{AM} = \Gamma(S)$ or the Black-Scholes inequality.

1.2 Pricing

Consider the transformation

$$u(x,\tau) = e^{\rho x + \xi t} V\left(S, T - \frac{2\tau}{\sigma^2}\right)$$

where

$$\begin{aligned} S &= Ke^x \\ t &= T - \frac{2\tau}{\sigma^2} \\ \rho &= -\left(\frac{r}{\sigma^2} - \frac{1}{2}\right) \\ \xi &= -\left(\frac{r}{\sigma^2} + \frac{1}{2}\right)^2 \end{aligned}$$

then we have

$$V(S,t) = v(x,\tau) = v\left(\log S, \frac{\sigma^2}{2}(T-t)\right)$$

[1]How?

[2]I'm not very clear on this...

With these substitutions we have shown that the Black-Scholes PDE transforms to the canonical heat equation $\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2}$ and so we have the similar inequality

$$-\frac{\partial u}{\partial t} + \frac{\partial^2 u}{\partial x^2} \leq 0$$

Then, with the transformed payoff function

$$g(x,t) = e^{\rho x + \xi t}\Gamma(S)$$

at (x,t) with $t > 0$ we get either

$$\begin{aligned} &u_t - u_{xx} > 0 \quad \text{and } u = g \text{ (i.e. the Black-Scholes inequality) or,} \\ &u_t - u_{xx} = 0 \quad \text{and } u > g \text{ (i.e. the Black-Scholes equation holds in equality)} \end{aligned}$$

with $u = g, u > g$ corresponding to $V_{AM} = \Gamma(S), V_{AM} > \Gamma(S)$, respectively. Then, we may multiply our two conditions into a single equation

$$\begin{aligned} (u-g)(u_t - u_{xx}) &= 0 \\ u_t - u_{xx} \geq 0, \quad u - g &\geq 0 \end{aligned}$$

If we consider the BTCS algorithm at a point (x_j, t_{n+1}) then we have

$$\begin{aligned} u_t = u_{xx} &\implies \frac{U_j^{n+1} - U_j^n}{\Delta t} = \frac{U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n-1}}{(\Delta x)^2} \\ &\implies U_j^{n+1} - U_j^n = \nu\left(U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1}\right) \quad \left(\text{with } \nu = \frac{\Delta t}{(\Delta x)^2}\right) \\ &\implies U_j^{n+1} = U_j^n + \nu\left(U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1}\right) \\ &\implies U_j^n = U_j^{n+1} - \nu\left(U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1}\right) \\ &\implies U_j^n = U_j^{n+1} - \nu U_{j+1}^{n+1} + 2\nu U_j^{n+1} - \nu U_{j-1}^{n+1} \\ &\implies U_j^n = -\nu U_{j+1}^{n+1} + (1+2\nu)U_j^{n+1} - \nu U_{j-1}^{n+1} \end{aligned}$$

Hence, under the BTCS algorithm, we have the conditions

$$\begin{aligned} (u-g)(u_t - u_{xx}) = 0 &\iff \left(U_j^{n+1} - g_j^{n+1}\right)\left(\nu U_{j+1}^{n+1} - (1+2\nu)U_j^{n+1} + \nu U_{j-1}^{n+1}\right) = 0 \\ u_t - u_{xx} \geq 0 &\iff U_j^n + \nu U_{j+1}^{n+1} - (1+2\nu)U_j^{n+1} + \nu U_{j-1}^{n+1} \geq 0 \\ u - g \geq 0 &\iff U_j^{n+1} - g_j^{n+1} - \geq 0 \end{aligned}$$

We specify this in matrix notation by

$$\begin{aligned} \left(\vec{U}_{n+1} - \vec{g}^{n+1}\right)^T\left(\mathbf{A}\vec{U}^{n+1} - \vec{b}\right) &= 0 \\ \mathbf{A}\vec{U}^{n+1} - \vec{b} &\geq 0 \\ \vec{U}^{n+1} - \vec{g}^{n+1} &\geq 0 \end{aligned}$$

where we mean $\vec{x} \geq 0$ to be that the the elements of $\vec{x}$ satisfy elementwise inequality with the elements of $\vec{0} = \{0, 0, ..., 0\}$. We are unable to just solve

$$\mathbf{A}\vec{U}^{n+1} - \vec{b} \geq 0$$

since the we may have that the second inequality $\vec{U}^{n+1} - \vec{g}^{n+1} \geq 0$ is not satisfied. So, we must solve the whole system simultaneously using some iterative method. We write

$$\vec{U}^{n+1} = \vec{g}^{n+1} + \vec{x}$$

We are looking for some vector $\vec{x}$ satisfying

$$\begin{aligned}
\vec{x}^T \left(\mathbf{A}\vec{x} - \hat{b}\right) &= 0 \\
\mathbf{A}\vec{x} - \hat{b} &\geq 0 \\
\vec{x} &\geq 0
\end{aligned}$$

where $\hat{b} = \vec{b} - \mathbf{A}\vec{g}^{n+1}$. To do so we split $\mathbf{A}$ into its lower, diagonal, and upper triangular component matrices

$$\begin{aligned}
\mathbf{A} &= \begin{bmatrix} 0 & 0 & \cdots & 0 \\ l_{2,1} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ l_{n,1} & l_{n,2} & \cdots & 0 \end{bmatrix} + \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{bmatrix} + \begin{bmatrix} 0 & u_{1,2} & \cdots & u_{1,n} \\ 0 & 0 & \cdots & u_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \\
&= \mathbf{L} + \mathbf{D} + \mathbf{U}
\end{aligned}$$

We write

$$\begin{aligned}
& (\mathbf{D} + \mathbf{L})\,\vec{x} \geq \hat{b} - \mathbf{U}\vec{x} \\
\implies & (\mathbf{D} + \mathbf{L})\,\vec{x} \geq \mathbf{D}\vec{x}\left[\hat{b} - (\mathbf{D} + \mathbf{U})\,\vec{x}\right]
\end{aligned}$$

Given some current guess of $\vec{x}$, say $\vec{x}^{(k)}$, we transform our task into an iterative problem and generate $\vec{x}^{(k+1)}$ by solving the inequality

$$\mathbf{D}\vec{x}^{(k+1)} \geq \mathbf{D}\vec{x}^{(k)} + \hat{b} - (\mathbf{D} + \mathbf{U})\,\vec{x}^{(k)} - \mathbf{L}\vec{x}^{(k+1)}$$

Note that this method requires $\mathbf{L}\vec{x}^{(k+1)}$ to be known to solve for the value at $k+1$, but by some matrix trickery we are able to know these values due to some property of lower triangular matrices & how we split up our matrix $\mathbf{A}$ into its components. We ensure that $\vec{x}^{(k+1)} \geq 0$ by setting

$$\vec{x}^{(k+1)} = \max\left[0, \vec{x}^{(k)} - \mathbf{D}^{-1}\left(\hat{b} + (\mathbf{D} + \mathbf{U})\,\vec{x}^{(k)} - \mathbf{L}\vec{x}^{(k+1)}\right)\right]$$

Since $\vec{x}^{(k+1)}$ appears on the RHS, if we freeze an iteration at any point midway through an update then the current solution vector is of the form

$$\left(x_1^{(k+1)}, ..., x_j^{(k+1)}, x_{j+1}^{(k)}, ..., x_{J-1}^{(k)}\right)$$

by the fact[3] that we have a lower triangular matrix $\mathbf{L}$

$$\begin{bmatrix} 0 & 0 & \cdots & 0 \\ l_{2,1} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ l_{n,1} & l_{n,2} & \cdots & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \vdots \\ x_{J-2}^{(k)} \\ x_{J-1}^{(k)} \end{bmatrix} = \begin{bmatrix} 0 & l_{2,1}x_2^{(k+1)} & \cdots \end{bmatrix}$$

Then we find

$$x_j^{(k+1)} = \max\left[0, x_k^{(k)} + \mathbf{A}^{-1}\left(\hat{b}_j - \mathbf{A}_{j,j}x_j^{(k)} - \mathbf{A}_{j,j+1}x_{j+1}^{(k)} - \mathbf{A}_{j,j-1}x_{j-1}^{(k+1)}\right)\right]$$

We have some parameter ω which may speed up[4] the problem

$$x_j^{(k+1)} = \max\left[0, x_k^{(k)} + \omega\mathbf{A}^{-1}(\cdots)\right]$$

So, we can rewrite the problem as finding a vector $\vec{x} \geq 0$ satisfying

$$\mathbf{A}\vec{x} - \hat{b} \geq 0$$

minimizing

$$G = \frac{1}{2}\vec{x}^T\mathbf{A}\vec{x} - \hat{b}^T\vec{x}$$

with stopping constraint

$$\vec{x}^{(k+1)} - \vec{x}^{(k)} < \vec{\epsilon}$$

The G is reduced at each stage as long as we have $\omega \in (0, 2)$. Note that this process was defined for an American put option. For an American call option we would never exercise an American call option early unless we have the option written on a stock with dividends.

2 Barrier Options

In some ways barrier options are more simple to consider than traditional (unbounded) options. The barrier specified in a given contract permits us to reduce the interval for which the asset price S may appears. That is, in a up & out call we have the value $V = 0$ if S ever goes above B. This permits us to "ignore" the region $S \geq B$ since we know the corresponding price.

[3] I think the point was that we have lots of multiplication by 0.

[4] Without motivation?

2.1 Up & Out Call Option

An up & out call option expires worthless if S passes barrier B before expiry T. That is,

$$V(B,t)=0 \quad t\in[0,T]$$

we have the boundary value problem (under Black-Scholes)

$$\begin{aligned} V_t + rSV_S + \frac{1}{2}\sigma^2 S^2 V_{SS} - rV = 0 \quad t\in[0,T), S\in[0,B] \\ V(0,t)=0 \\ V(S,t)=\Gamma(S) \quad S\in(0,B) \end{aligned}$$

We may still need to specify some artificial boundary for S near 0, say $0+\epsilon=\epsilon$, but we have some natural for $S\to\infty \implies B$ as a boundary.

2.2 Up & In Option

If we wanted to price an up & in option it is often times more single to use the decomposition

$$(\text{Up \& In}) + (\text{Up \& Out}) = \text{Vanilla}$$

where we are able to price the appropriate up & out and vanilla options.

3 Lookback Options

3.1 Floating Strike Lookback Put

In the case of a floating strike lookback put we have payoff function

$$\Gamma(S)=(M-S)^+$$

where

$$M=\max_{t\in[0,T]} S_t$$

That is, we have the floating strike M as the max price which the asset S attains over the life of the option. We write

$$M(t)=\max_{\tau\in[0,t)} S(\tau)$$

where $M(t)$ is the max price the asset S attains up to t. Then we see that the option value is clearly a function of S, M, and t is the region $0\le S\le M$. Hence,

$$V=V(S,M,t)$$

when we have $S(t)<M(t)$ (i.e. the price S has decreased from its max) then we have $M(t)$ is not changing in time. If $M(t)=M$ constant, we have that V must solve the traditional

Black-Scholes equation. When $S(t) = M(t)$ (i.e. the floating strike is at the current price of the asset), for $0 < t < T$, we note

$$\mathbb{P}\left(M(t) = M(T)\right) = 0 \quad \text{(i.e. the floating strike will change with prob. 1)}$$

Thus, we find that the option value does not depend on $M(t)$[5] implying

$$\frac{\partial V}{\partial M} = 0 \quad \text{when } S(t) = M(t)$$

and so at $t = T$ we have option value

$$\Gamma(S, M) = (M - S)^+$$

We end up with a system of PDEs

$$\begin{aligned}
&V_t + SV_S + \frac{1}{2}\sigma^2 V_{SS} = rV \quad t \in [0, T),\ S \in [0, M],\ M > 0 \\
&\begin{cases} V(0, M, t) \approx Me^{-r(T-t)} & t \in [0, T) \\ V(S, M, T) = \Gamma(S, M) & S \in [0, M],\ M > 0 \end{cases} \\
&\frac{\partial V}{\partial M}(M, M, t) = 0 \quad t \in [0, T),\ M > 0
\end{aligned}$$

We reduce the dimension of the problem by scaling with $\frac{S}{M} = \xi$. Let

$$\begin{aligned}
\Gamma(S, M) &= M(1 - \xi)^+ \\
W(\xi, t) &= \frac{1}{M} V(S, M, t) \\
P(\xi) &= \frac{1}{M} \Gamma(S, M)
\end{aligned}$$

Then, for $\xi \in (0, 1)$ we have

$$\begin{aligned}
W_t + \frac{1}{2}\sigma^2 \xi^2 W_{\xi\xi} + r\xi W_\xi - rW &= 0 \\
W(\xi, T) = P(\xi) &= (1 - \xi)^+
\end{aligned}$$

But we have the boundary condition $S = M \iff \xi = 1$, so

$$\begin{aligned}
0 = \frac{\partial V}{\partial M} &= \frac{\partial}{\partial M} MW(\xi, t) \\
&= W + M \frac{\partial W}{\partial \xi} \frac{\partial \xi}{\partial M} \\
&= W + M \frac{\partial W}{\partial \xi} \frac{\partial}{\partial M} \left(\frac{S}{M} \right) \\
&= W + M \frac{\partial W}{\partial \xi} \left(-\frac{S}{M^2} \right) \\
&= W - \xi \frac{\partial W}{\partial \xi}
\end{aligned}$$

[5] I'm still not convinced of this...

So, on the line $\frac{S}{M} = \xi = 1$ we have

$$0 = W - \xi W_\xi$$

and for a put at $\frac{S}{M} = \xi = 0$ we have

$$W(0,t) = \frac{1}{M}V(0,M,t) \approx \frac{1}{M}Me^{-r(T-t)} = e^{-r(T-t)}$$

Setting $x = \log \xi \iff \xi = e^x$ and

$$u(x,t) = e^{\rho x + \xi t} W\left(e^x, T - \frac{2t}{\sigma^2}\right)$$

we see that the right boundary is at $x = 0$ and

$$(\rho + 1)u = u_x$$

We may approximate the boundary numerically via a finite difference algorithm by

$$(\rho + x)U_J^n = \frac{U_J^n - U_{J-1}^n}{\Delta x}$$

so (on the boundary) we get

$$U_J^n = \frac{1}{1 - (\rho + 1)\Delta x} U_{J-1}^n$$

and we may transform u back to V to get our final option value.

3.2 Fixed Strike Lookback Put

We have payoff with fixed strike K

$$\Gamma(S, M) = (K - M)^+$$

It turns out that no dimension reduction is possible and so we must solve the coupled PDEs directly.

4 Final Notes on PDE Methods

A major issue with PDE methods is that a particular problem will require a usually unique implementation that is specific to the features of the problem.

The main PDE problems to remember are

1. European options (as a baseline)
2. American options
3. Any path dependent options