MATHEMATICAL FINANCE
An International Journal of Mathematics, Statistics and Financial Economics

Mathematical Finance, Vol. 24, No. 3 (July 2014), 505–532

BOUNDARY EVOLUTION EQUATIONS FOR AMERICAN OPTIONS

DANIEL MITCHELL

McCombs School of Business, University of Texas at Austin

JONATHAN GOODMAN

Courant Institute of Mathematical Sciences, New York University

KUMAR MUTHURAMAN

McCombs School of Business, University of Texas at Austin

We consider the problem of finding optimal exercise policies for American options, both under constant and stochastic volatility settings. Rather than work with the usual equations that characterize the price exclusively, we derive and use boundary evolution equations that characterize the evolution of the optimal exercise boundary. Using these boundary evolution equations we show how one can construct very efficient computational methods for pricing American options that avoid common sources of error. First, we detail a methodology for standard static grids and then describe an improvement that defines a grid that evolves dynamically while solving the problem. When integral representations are available, as in the Black–Scholes setting, we also describe a modified integral method that leverages on the representation to solve the boundary evolution equations. Finally we compare runtime and accuracy to other popular numerical methods. The ideas and methodology presented herein can easily be extended to other optimal stopping problems.

KEY WORDS: optimal stopping, American options, stochastic volatility, early exercise boundary, free-boundary problem, dynamic grid.

1. INTRODUCTION

American style options provide the holder the right to trade an underlying asset for a specified strike price before a specified expiry time. Pricing and finding the optimal exercise policy, which is known to be a surface that partitions the domain into exercise and hold regions, are interrelated and are solved for by transforming them to differential equation problems. The resulting differential equation, along with boundary conditions, formulate a free-boundary problem and characterize the price of the option. An accurate computation of the solution to the free-boundary problem relies on an accurate representation of the boundary and an accurate treatment of its dynamics. Rather than work with the equation that characterizes the price evolution of an American security exclusively, one could potentially derive and use the equations that characterize the evolution of the free-boundary for computational purposes. This however has not been seen as a valuable

Manuscript received March 2011; final revision received February 2012.
Address correspondence to Daniel Mitchell, McCombs School of Business, University of Texas at Austin, Austin, TX 78712, USA; e-mail: daniel.mitchell@phd.mccombs.utexas.edu.

DOI: 10.1111/mafi.12002

method because the boundary evolution equation also depends implicitly on the price, which seemed to be challenging to handle efficiently.

In this paper we consider American options, in the Black–Scholes setting and in a stochastic volatility setting, and derive boundary evolution equations. We show how one can construct computational procedures that efficiently utilize these evolution equations to compute both the price and the optimal exercise policy of American options. The evolution equations tell us exactly how fast the exercise boundary should move in time. This speed is dependent on both the current level of the boundary and a mixed derivative of the price function at the boundary, resulting in a system of differential equations. By solving these equations simultaneously we can track both the optimal exercise policy and the price function.

A challenge in constructing a boundary evolution equation–based computational procedure, apart from that posed by the implicit dependence of the equation on the price, is in taming the errors that arise from having to choose amongst points on a predefined grid to represent the boundary. Hence, we first construct a computational procedure that works on the standard rectangular cartesian grid by allowing a boundary to float between grid points. Although the performance of this first step is very encouraging, one could potentially eliminate any error due to the grid and boundary mismatch by allowing the grid to adapt to the boundary rather than predefine it. To this extent we next construct an improved methodology that dynamically builds a nonlinear grid while solving the boundary evolution equations. Such a dynamic evolution while being a relatively complicated implementation, performs significantly better and becomes essential under stochastic volatility. For cases where an integral representation of the option price is available, as is the case for the Black–Scholes model, we could potentially use the representation for further efficiency in solving the boundary evolution equation. We demonstrate how this can be done too for the Black–Scholes case. We also provide numerical evidence that the methods constructed in this paper are faster and more accurate than other relevant numerical methods. More work could potentially be done to extend this to existing integral representations of American options with stochastic volatility.

The primary objective of the paper is to show that it is possible to construct efficient numerical methods that take advantage of boundary evolution equations to avoid common sources of error. The expressions for the dynamics of the boundary in the Black–Scholes setting was found in van Moerbeke (1975) then rediscovered, independently, in Goodman and Ostrov (2002), and they were extended to some multifactor models in Hayes (2006). The corresponding equations for the stochastic volatility case have been derived in this paper. American option pricing is probably the most popular example amongst a larger class of very similar problems known as optimal stopping problems. Although this paper focuses exclusively on American options, the arguments for deriving the boundary evolution equations and the computational methods that solve these equations can readily be extended to other optimal stopping problems.

In Section 2, we consider the classical Black–Scholes setting of constant volatility and present three methods which leverage on the boundary equation. We then compare these methods to other numerical methods and find that all three methods constructed perform better. Section 3 considers the stochastic volatility case with a setting that is general enough to encompass several popular stochastic volatility models. For numerical comparisons, we only consider the most popular Heston model and describe the dynamic grid-based method in this context. We then compare the method to other existing numerical methods and find improved performance.

1.1. Background and Previous Literature

A put option is a contingent claim that gives the owner the right, but not the obligation, to sell a share of stock (or any other asset) at a prespecified price. Throughout the paper we restrict discussion to the put option only because almost the same arguments and equations will hold for the call option, wherein the owner has the right to buy a share of stock at a prespecified price. Put options come in two main flavors, "European," where the owner can only exercise this right at one prespecified time (expiration), and "American," where the owner can exercise this right anytime before expiration. When valuing an American put option the crucial step is to find the optimal early exercise boundary, which indicates the circumstances under which the option should be exercised before it expires. While a closed form solution for the value of a European option with constant volatility was found in the classical paper by Black and Scholes (1973) and for one particular stochastic volatility model in the paper by Heston (1993), there is no known closed form solution for the value of an American option with constant or stochastic volatility.

For constant volatility there are two main classes of numerical methods that approximate the price of American options. The first class computes the expected value of the American's payoff under the risk neutral measure. This class usually consists of Monte Carlo and binomial methods, and these methods only find the price of the option for one particular price and time to expiration and are typically unable to compute the early exercise boundary efficiently. The second class rephrases the expected value as the solution to a free-boundary partial differential equation (PDE), and methods in this class find the entire pricing function and the early exercise boundary. It can be difficult to compare methods in different classes because PDE methods give much more information than the first class.

The most well-known numerical method for solving the free-boundary problem was developed by Brennan and Schwartz (1977), but there have been several numerical methods developed since then. Muthuraman (2008) uses an iterative method to convert the free-boundary problem into a sequence of fixed boundary problems. Also Goodman and Ostrov (2002) find a differential equation that governs the early exercise boundary, which will be used heavily in this paper, and use it to derive a short-time asymptotic expansion of the boundary.

There are other methods that do not solve the free-boundary problem; rather they evaluate the risk neutral expected value of the option's payoff. Two common methods in practice that solve this problem are the binomial and trinomial tree methods. The binomial method was first seen in Cox, Ross, and Rubinstein (1979). Also, there has been much success in solving this problem using Monte Carlo simulation, most notably by Tilley (1993), Broadie and Glasserman (1997), and Longstaff and Schwartz (2001). Other methods that solve this problem partition the price as a European option's price plus an early exercise premium which results in an integral equation (Kim 1990; Jacka 1991; Carr, Jarrow, and Myneni 1992).

Recently there has been some work that exploits asymptotic analysis of the early exercise boundary to find approximate closed form solutions to the American option problem in the Black–Scholes setting. In Bunch and Johnson (2000) the authors find an implicit equation that can approximate the boundary at any time and then use numerical integration to find the price of the option. In Stamicar, Sevcovic, and Chadam (1999) the authors find an approximate explicit formula for the early exercise boundary. In Chen and Chadam (2007) the authors provide a detailed mathematical analysis of the early

exercise boundary and provide an implicit ordinary differential equation (ODE) that governs the boundary. Evans, Kuske, and Keller (2002) provide results for American options on dividend paying stocks. A more comprehensive comparison of numerical methods can be found in AitSahlia and Carr (1997).

In the years since the seminal work of Black and Scholes there have been many empirical studies that suggest that simple Geometric Brownian Motion does not capture enough of the dynamics of a stock price to give an accurate price for derivative securities. As a result people have studied the case when the volatility of the stock follows a stochastic process. There have been several models that incorporate this but most work has focused on European options. As in the Black–Scholes setting, there is no known closed form solution for American options under any model.

Despite the vast research in American options with constant volatility there has been relatively less work exploring stochastic volatility. While some of the methods mentioned above can be extended to handle stochastic volatility, namely the PDE and Monte Carlo methods, there are also many methods that cannot handle stochastic volatility. Given the limitations of the above methods there has been some work looking for fast methods to price American options with stochastic volatility, including the multigrid method in Clarke and Parrott (1999) and a moving boundary method by Chockalingam and Muthuraman (2010). Ikonen and Toivanen (2007) use a componentwise splitting method to create three simple linear complementarity problems which they solve using the Brennan–Schwartz method. Also Wilmott (1998) describes how to use projected successive over relaxation (PSOR) to solve the free-boundary problem. In Detemple and Tian (2002) the authors present an integral representation for American options with stochastic volatility and interest rates that can be recursively solved to find the early exercise boundary. Broadie et al. (2000) use nonparametric techniques to investigate properties of the early exercise boundary under stochastic dividends and volatility. In Ikonen and Toivanen (2008) the authors present a more exhaustive review of other computational methods for American options with stochastic volatility.

2. CONSTANT VOLATILITY

In this section we generalize a boundary evolution equation, for the Black–Scholes setting, found in Goodman and Ostrov (2002), to a more general setting than nondividend paying stocks that includes assets such as futures, dividend paying stocks, and options on foreign currency. We then develop three numerical methods that leverage on the boundary evolution equation to obtain fast and accurate approximations of the price of an American option with constant volatility. For the rest of the paper we use the notation of Karatzas and Schreve (1998).

2.1. The Boundary Equation

We start with the classical Black–Scholes PDE for valuing an American put option, $p(x, \tau)$, where x is the price of the underlying asset and τ is the time until expiry. An American put option can be exercised at any time before it expires with payoff of $q - x$, where q is the strike price of the option. This suggests that we should partition the domain into two distinct regions separated by the early exercise boundary, $c(\tau)$. If at time τ, $x \leq c(\tau)$ the option should be exercised immediately with a payoff of $q - x$, and if $x > c(\tau)$ the option should be held. The optimal choice of $c(\tau)$ is decided by

comparing the intrinsic value of the option to its tradable value; if it is worth more on the open market than its intrinsic value, then it should not be exercised. In the constant volatility case if $x > c(\tau)$ then $p(x, \tau)$ is governed by the classical Black–Scholes PDE,

$$\frac{\partial p}{\partial \tau} = \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 p}{\partial x^2} + bx\frac{\partial p}{\partial x} - rp. \tag{2.1}$$

Here r is the risk-free interest rate, σ is the volatility of the underlying asset, and b is the instantaneous cost of carrying the underlying asset, as in Huang, Subrahmanyam, and Yu (1996). Using this notation for b allows us to price several financial instruments. For example, for nondividend paying stocks, $b = r$; for stocks with constant dividend yield δ, $b = r - \delta$; for futures, $b = 0$; and for options on foreign currency with foreign risk-free rate r_f, $b = r - r_f$.

We know that at $\tau = 0$ the option expires, thus it must be exercised or abandoned and therefore $c(0) = q$, if $b \geq 0$, otherwise $c(0) = \frac{r}{r-b}q$. The last thing we need to know about this option is the smooth pasting condition, which states that on the boundary p must be differentiable as shown in Merton (1992). With this information we can establish initial and boundary conditions for p, which are

$$p(x, 0) = \max(q - x, 0),$$

$$p(c(\tau), \tau) = q - c(\tau), \tag{2.2}$$

$$\frac{\partial p(c(\tau), \tau)}{\partial x} = -1, \tag{2.3}$$

and

$$\lim_{x\to\infty} p(x, \tau) = 0. \tag{2.4}$$

Equation (2.4) implies that

$$\lim_{x\to\infty} \frac{\partial p}{\partial x} = 0, \tag{2.5}$$

because p is convex and decreasing, as seen in Karatzas and Shreve (1998). It is more convenient, numerically, to use equation (2.5) as a boundary condition for large x, so we will not use equation (2.4) in numerical experiments.

Now that we have the boundary conditions we would like a differential equation that governs $c(\tau)$. We find this using higher-order derivatives that are continuous up to the early exercise boundary from the continuation region, but not across into the exercise region (see, for example, Lawrence and Salsa 2008 for a proof of this in several multiasset cases). We use these expressions to treat the price and boundary as a coupled system to be solved simultaneously.

THEOREM 2.1. *The differential equation that governs $c(\tau)$ is*

$$\frac{\partial c(\tau)}{\partial \tau} = -\frac{\partial^2 p(c(\tau), \tau)}{\partial x \partial \tau} \frac{\sigma^2 c^2(\tau)}{2qr - 2(r - b)c(\tau)}. \tag{2.6}$$

Proof. First differentiate the boundary conditions with respect to time, which will be in terms of the time derivative of p. This will also lead to time derivatives of the exercise boundary, $c(\tau)$, which is indeed differentiable by lemma 4.1 in Myneni (1992). Notice

that the time derivative of p also satisfies equation (2.1). The first boundary condition, (2.2) becomes

$$\frac{\partial p}{\partial x}\frac{\partial c}{\partial \tau} + \frac{\partial p}{\partial \tau} = -\frac{\partial c}{\partial \tau}$$

which is simplified using equation (2.3) to

$$\frac{\partial p(c(\tau), \tau)}{\partial \tau} = 0. \tag{2.7}$$

Next take the time derivative of equation (2.3) and find

$$\frac{\partial^2 p}{\partial x^2}\frac{\partial c}{\partial \tau} + \frac{\partial^2 p}{\partial x \partial \tau} = 0. \tag{2.8}$$

Now take the limit as $x \to c(\tau)$ from the right and substitute equation (2.7) into equation (2.1) and get

$$0 = \frac{1}{2}\sigma^2 c^2(\tau)\frac{\partial^2 p(c(\tau), \tau)}{\partial x^2} + bc\frac{\partial p(c(\tau), \tau)}{\partial x} - rp(c(\tau), \tau). \tag{2.9}$$

Next substitute equations (2.2) and (2.3) into (2.9),

$$\frac{\partial^2 p(c(\tau), \tau)}{\partial x^2} = \frac{2qr - 2(r-b)c(\tau)}{\sigma^2 c^2(\tau)}. \tag{2.10}$$

Finally combine equations (2.8) and (2.10) and rearrange terms to find the desired function, (2.6). □

Even with this equation, finding the price of the American put is still a hard problem. We see in equation (2.6) that the boundary's evolution depends on both a mixed derivative of the price function and the current boundary level, which creates a system of nonlinear differential equations. The price of the put option depends on the boundary and the boundary depends on the price of the put. To solve these equations we must find a way to evolve them simultaneously.

Figure 2.1 shows the state space partitioned into the exercise region and the continuation region. The two regions are separated by the early exercise boundary. In the exercise region the price of the put is equal to its intrinsic value. In the continuation region the price of the put is governed by equation (2.1).

2.2. Numerical Method on a Static Grid

This section constructs a numerical method that uses equation (2.6) to compute the early exercise boundary and the price function of an American put option. The basic idea is to step forward in time to expiry discretely, evolving p and c at each step using finite difference approximations to equations (2.1) and (2.6). In this process several intricacies need to be addressed so we describe the algorithm with a three-step iterative procedure.

Step 1: Initialize p and c at a small time before expiration. To begin evolution using equations (2.1) and (2.6) we need an initial value of p and c. We know that at $\tau = 0$ the boundary is located at $c(0) = \min(q, \frac{r}{r-b}q)$, and at every value of x such that $x \geq q$ we have $p(x, 0) = 0$. In this numerical method we only consider the domain where $x \geq c(\tau)$

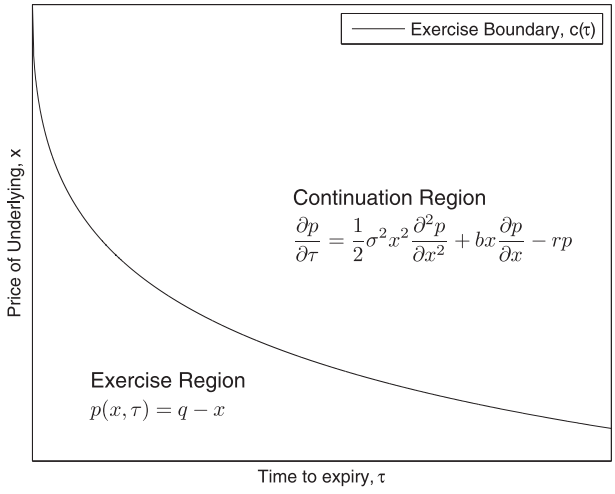

FIGURE 2.1. Partitioned state space.

because when $x > c(\tau)$ we know that p is governed by equation (2.1) and when $x = c(\tau)$, p must obey the boundary conditions so any value of $x < c(\tau)$ cannot be used. Using this information as an initial value we will see that all finite difference approximations to derivatives in the x variable will be zero if $b \geq 0$. This happens because we do not consider the domain such that $x < c(0)$, the only place where $p(x, 0) \neq 0$, so any place that we calculate a derivative in x will result in a linear combination of zeros, which is zero. For example, if $q = 100$ and $b > 0$ then $c(0) = q$. Now if we try to approximate the derivative of p at $x = 101$, using a finite difference approximation, we will find that $p(101 + \Delta x, 0) = p(101 - \Delta x, 0) = 0$, and $\frac{\partial p}{\partial x} \approx \frac{p(101+\Delta x,0)-p(101-\Delta x,0)}{2\Delta x} = \frac{0-0}{2\Delta x} = 0$.

This fact together with equation (2.1) tells us also that the numerical approximation of $\frac{\partial p}{\partial \tau}$ must also be zero. This means the price of the put cannot change in one step, and thus the location of the boundary cannot change in one step, if we start with the initial data at $\tau = 0$. If they do not move in the first step then they will not move in any subsequent step and the price of the put will stay at zero for all times to expiry, which is clearly incorrect. However, if $b < 0$ we do not have a problem.

To overcome this problem we approximate the put, at a short time before expiration, as a European option, as in Broadie and Detemple (1996), and with this we can take advantage of the closed form Black–Scholes equation for European options. On a discrete set of equally spaced grid points in x between zero and some $\hat{x}$, where $\hat{x}$ is the maximal value in the computational domain, we find the value of a European put, $f(x, \tau_o)$, a short time before expiration, τ_o. The choice of $\hat{x}$ is not entirely trivial here, we need to pick $\hat{x}$ so that equation (2.5) is approximately true for all times to expiry that we consider. Since f is a European option it must satisfy $f(x, \tau_o) = qe^{-r\tau_o}N(-d_2) - xe^{-(r-b)\tau_o}N(-d_1)$, where $d_1 = \frac{\log(x/q)+(b+\frac{1}{2}\sigma^2)\tau_o}{\sigma\sqrt{\tau_o}}$ and $d_2 = d_1 - \sigma\sqrt{\tau_o}$. Here N is the standard normal cumulative distribution function. We initialize p for a short time as $p(x, \tau_o) = \max(f(x, \tau_o), q - x)$.

To find the initial value for the boundary, $c(\tau_o)$, we use a binary search to find the place where $f(x, \tau_o)$ intersects the line $q - x$. Here we will almost certainly find that $c(\tau_o)$ is not located at one of the grid points chosen above, but this is not a problem; we will evolve p on the fixed grid and let c move between the grid points.

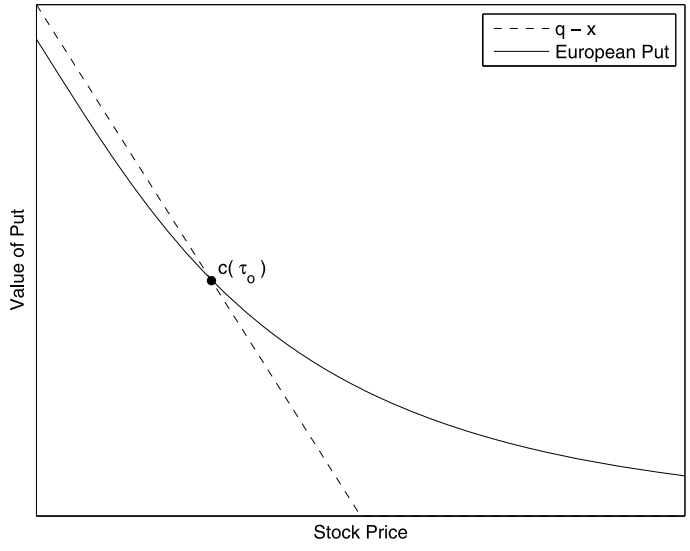

FIGURE 2.2. Initialization of $p(x, \tau_o)$ and $c(\tau_o)$.

In Figure 2.2, we illustrate how to initialize p and c. In the figure the dashed line represents the intrinsic value of the put and the solid line represents the value of a European option. We say that to the left of the intersection the American is equal to the dashed line, to the right of the intersection the American is equal to the solid line and the boundary is located at the intersection of the two lines. However we only work in the domain such that $x \geq c(\tau)$ so we only need the location of the intersection and the solid line to the right for initialization. Figure 2.2 exaggerates the initialization procedure for illustrative purposes. In the numerical experiments we run in Section 2.5 we find that the slope of the European option at the initial approximate boundary ranges between -0.999 and -0.975 when we initialize at half a trading day before expiration, $\tau_0 = \frac{1}{2}\frac{1}{252}$.

Step 2: Evolve p one step in time to expiry and approximate the mixed derivative. The next step is to evolve p one step backwards in time, holding $c(\tau)$ fixed. This, however, presents a problem because the values of p are not exactly uniform and we want to use a finite difference method. The grid points where we know p are uniformly spaced, but we also know the value of p at the boundary, which does not fit on this uniform spacing. To use a finite difference method we need to approximate all derivatives using the value of the price function at discrete grid points. For most of the grid points we can use standard central difference methods, however at the first grid point to the right of the boundary, call this point x_0, we cannot use these standard central difference formulae. To find the derivatives of p at x_0 we use Taylor series expansion to derive noncentral finite difference approximations involving x_0, $x_0 + h$ and $x_0 - h_2$. Here h_2 is the distance between x_0 and $c(\tau)$, and $p(x_0 - h_2) = q - c(\tau)$ because $x_0 - h_2 = c(\tau)$. One advantage of using this method to compute the derivatives of p at x_0 is that we can insert these equations directly into any discrete time stepping finite difference algorithm, like the Crank and Nicolson (1947) algorithm, which we use.

In the evolution of p we do not need to calculate any derivatives at $c(\tau)$ or $\hat{x}$ because we can use equations (2.2) and (2.5) as boundary conditions. Equation (2.2) means that in one time step the value of the put at $c(\tau)$ does not change. Equation (2.5) means that the

value of the put at $\hat{x}$ is equal to the value of the put at the grid point just before $\hat{x}$. Both of these boundary conditions can easily be satisfied implicitly using the Crank–Nicolson algorithm.

After we evolve p we need to approximate the x derivative of the price function at the early exercise boundary so that we can use it to calculate $\frac{\partial^2 p(c(\tau),\tau)}{\partial x \partial \tau}$. To calculate this derivative we need to use the location of the boundary, $c(\tau)$, the value of the put at the boundary, $q - c(\tau)$, and a few grid points to the right of the boundary. In this calculation we cannot simply use standard one-sided finite difference formulae because the boundary is not located at a grid point. This means that the places where we know the value of the put to the right of the boundary are not equally spaced; if the distance between grid points is h then the space between the boundary and the first grid point to the right of the boundary must be less than h. To overcome this problem we fit a spline through the boundary and a few grid points to the right of the boundary. Using the coefficients of this spline we can analytically approximate the derivative of the price function at the boundary.

With this value for the x derivative we can approximate $\frac{\partial^2 p(c(\tau),\tau)}{\partial x \partial \tau}$ using a first-order finite difference method in time. If we say the value of the x derivative at the boundary before we evolved p is $p_x^{\text{old}} = -1$ and the value after we evolved p is p_x^{new} then $\frac{\partial^2 p(c(\tau),\tau)}{\partial x \partial \tau} \approx \frac{p_x^{\text{new}}+1}{\Delta\tau}$, where $\Delta\tau$ is the step size in time to expiry.

Step 3: Evolve c one step in time to expiry. Now that we have evolved p and calculated the mixed derivative at the boundary we need to evolve $c(\tau)$ one step in time to expiry to catch up with p. For this we hold $\frac{\partial^2 p(c(\tau),\tau)}{\partial x \partial \tau}$ fixed and use equation (2.6) and an explicit Runge–Kutta method to evolve c, we use the second-order Runge–Kutta method in numerical experiments; see Iserles (2008) for details on Runge–Kutta methods.

There is one last problem we face: what happens when the boundary crosses from one side of a grid point to the other? For example, if at time τ, $c(\tau)$ is located between the 99th and 100th grid points and at time $\tau + \Delta\tau$, $c(\tau)$ is located between the 98th and 99th grid points, what do we do? Here there are a few options as well. In Chen et al. (1997) the authors suggest using a spline to interpolate the value of p, however we find that simply leaving p at this point as its intrinsic value, $q - x$, does not lead to any significant reduction in accuracy, and thus using a spline is not worth the added complexity. We then repeat Steps 2 and 3 until we reach the desired time before expiration.

2.3. Numerical Method on a Dynamic Grid

In the previous section we allowed the optimal early exercise boundary to move between fixed grid points in the computational domain, however this can lead to some error when the boundary is very close to the next grid point and h_2 is very small when compared to h. To overcome this error we use numerical grid generation to force the grid points to conform to the boundary at every time step by using a change of variables. This forces the grid points we use, to approximate p, to move over time, and the space between the boundary and the closest grid point remains a constant. This has the advantage of allowing us to use standard high-order finite difference approximations when calculating the mixed derivative at the boundary and the standard difference methods at the first grid point greater than the boundary.

Numerical grid generation is used to transform complicated computational domains, through a change of variables, to much simpler domains that allow the use of standard

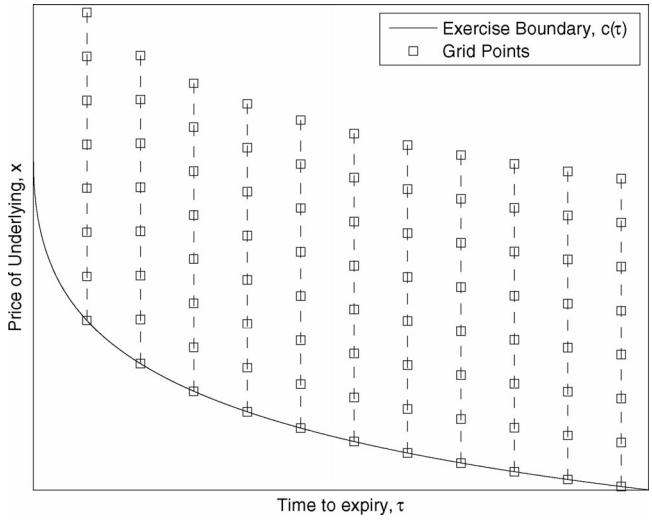

FIGURE 2.3. Computational grid in x at different times τ.

finite difference methods. In our case we wish to transform the domain $\{(x, \tau)\colon \forall\tau \geq 0, x \geq c(\tau)\}$ to $\mathbb{R}^2_+$. For details of numerical grid generation see Thompson, Warsi, and Mastin (1985). The front fixing methods in Nielsen, Skavhaug, and Tveito (2002) and Wu and Kwok (1997) also use a change of variables to eliminate the moving boundary, however this does not translate to a computational advantage because they do not use the boundary evolution equation considered here. The change of variable we use to transform our domain is

$$\omega = x - c(\tau), \tag{2.11}$$
$$g(\omega, \tau) = p(x, \tau).$$

Here ω can be interpreted as distance to the boundary. Given this change in variable we discretize ω uniformly from zero to some $\hat{\omega}$, which is equivalent to re-discretizing x at every time step uniformly from $c(\tau)$ to some $\hat{x}$, where $\hat{x}$ changes at each step to $c(\tau) + \hat{\omega}$. Figure 2.3 shows the computational grid in the (x, p) space for different values of τ. As τ increases, $c(\tau)$ decreases and the grid points align with the boundary for every value of τ. After transformation to the (ω, g) space the computational grid is a standard rectangular region.

Using the chain rule we find the PDE that governs the evolution of g and c to be

$$\frac{\partial g}{\partial \tau} = \frac{1}{2}\sigma^2(\omega + c(\tau))^2 \frac{\partial^2 g}{\partial \omega^2} + b(\omega + c(\tau))\frac{\partial g}{\partial \omega} - rg + \frac{\partial g}{\partial \omega}\frac{\partial c}{\partial \tau}, \tag{2.12}$$

$$\frac{\partial c(\tau)}{\partial \tau} = -\frac{\partial^2 g(0, \tau)}{\partial \omega \partial \tau}\frac{\sigma^2 c^2(\tau)}{2qr - 2(r - b)c(\tau)}. \tag{2.13}$$

The main difference between equations (2.12) and (2.1) is the addition of the final nonlinear term which comes from using the chain rule to differentiate g with respect to time. Known as the grid speed, this term allows us to find the value of p at the new grid points without the need for any sort of interpolation. Here, however, it is not as

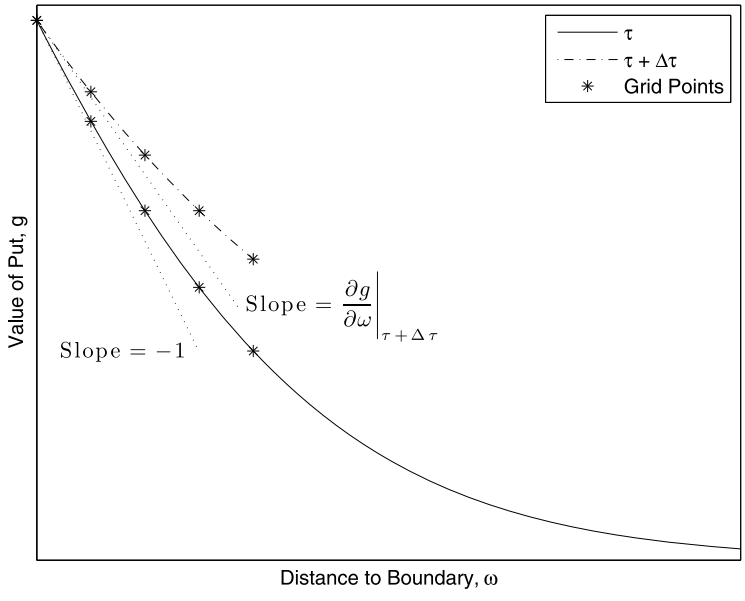

FIGURE 2.4. Difference in slopes, to calculate $\frac{\partial^2 g}{\partial \omega \partial \tau}$.

easy to estimate the mixed derivative at the boundary and we must come up with a new method of approximation. As the grid speed term in equation (2.12) also depends on the boundary evolution equation we cannot simply evolve g one step and use that to calculate the boundary evolution. The numerical method presented here can be described by a three-step iterative procedure as well.

Step 1: Initialization. We initialize p for a small time before expiry, τ_0, the same way we did in Section 2.2. However in this case we first find $c(\tau_0)$ and then initialize p uniformly between $c(\tau_0)$ and $\hat{x}$. Then we assign these values to ω and g according to the change of variables (2.11).

Step 2: Calculate $\frac{\partial^2 g(0,\tau)}{\partial \omega \partial \tau}$. Here, again, the calculation of the mixed derivative at the boundary can be difficult. To approximate this we simply evolve a few grid points greater than the boundary using equation (2.12) without the last term, the grid speed, and calculate the ω derivative after this evolution to use in a time finite difference method. Omitting the grid speed term has the effect of freezing the grid points for one small step in time and telling us how much the price of the put would change on those fixed grid points. For example if we use a 4-point finite difference approximation for the ω derivative and a second-order Runge–Kutta method for the time derivative, we only need to evolve six grid points larger than the boundary, so that we do not need to worry about right boundary conditions, which is not computationally expensive so we use this in numerical experiments.

Figure 2.4 shows how we calculate $\frac{\partial^2 g(0,\tau)}{\partial \omega \partial \tau}$ in the (ω, g) space. We see that all grid points are equally spaced and the grid points on the solid and dashed lines correspond to the same horizontal values because we dropped the grid speed term to get the dashed line. We also see that we only have the value of the put on the dashed line for a few grid points. Using the grid points on the dashed line and a standard one-sided finite difference

equation we calculate the ω derivative. Using these two values with equation (2.3) we can approximate $\frac{\partial^2 g(0,\tau)}{\partial\omega\partial\tau}$.

Step 3: Evolve g and c simultaneously one step. Once we have calculated the mixed derivative we hold it constant while we evolve equations (2.12) and (2.13). Since we hold this constant we can use a coupled Runge–Kutta method to evolve g and c; in numerical experiments we use the second-order coupled Runge–Kutta method. Also for this method we use the same boundary condition for large ω as we did in Section 2.2. However to increase accuracy we add an extra grid point to the end of the computational domain every time $c(\tau) + \hat{\omega} < \hat{x}$. When this extra grid point is brought into the computational domain it is introduced according to equation (2.5). We repeat Steps 2 and 3 until we reach the desired time and then change the variables back to x and p.

We could potentially evolve the system implicitly but equations (2.12) and (2.13) are both nonlinear. This means we either need to linearize the equations or use a nonlinear solver to evolve the system, however we do not want to rely on the speed of any specific nonlinear solver to determine the computational time of the algorithm. When we consider stochastic volatility we will have nonlinear PDEs similar to these and we apply a linearization to the system. However in constant volatility the linearization is not beneficial on a fine mesh so we only use an explicit method.

In this method we cannot evolve g one step, calculate the mixed derivative, and then evolve c one step as we did in the previous section because the grid is moving. Considering the (x, p) space in the evolution of the price with the grid speed term, the grid points used to calculate the space derivative before the price evolution and the grid points used after the evolution are not the same. Therefore we cannot combine these values to calculate the mixed derivative. Also, the value of the grid speed term is partially determined by the mixed derivative. If we do not know the value of the mixed derivative, then we do not know the value of the grid speed term and we cannot evolve g through time.

Figure 2.5 shows the evolution of the value of the put and the grid points used to calculate its value over a step in time to expiry. The solid line represents the value of the put in the (x, p) space and the corresponding grid points in x before the price and boundary are evolved one step. The dashed line shows the value of the put and the corresponding grid points after the price and boundary are evolved in Step 3. We can see that the grid points on the solid and dashed curves do not coincide because they have moved over the course of a step in time to expiry. The first grid point on each line corresponds to the early exercise boundary at that time.

There are a couple of minor drawbacks to this algorithm. The first problem is that we will almost always have to use a spline to compute the price of the put at some x value after the algorithm is finished because we cannot pick the grid to include that value like we could in the method presented in Section 2.2. For example, if we wish to know the price of the option when the underlying stock costs \$100 the method in Section 2.2 lets us pick 100 to be a grid point since the grid is static. However if we choose the grid spacing so that 100 is a grid point when we initialize we will almost certainly find that 100 is not a grid point when the algorithm is finished because the grid has moved. Therefore, we must interpolate to find the value of the option when the underlying costs \$100. The next problem is that since we include the nonlinear term to the end of equation (2.12) the CFL condition forces us to use a smaller time step size than that required in Section 2.2. We will see however that despite these problems this method compares favorably in speed and accuracy to the static grid method.

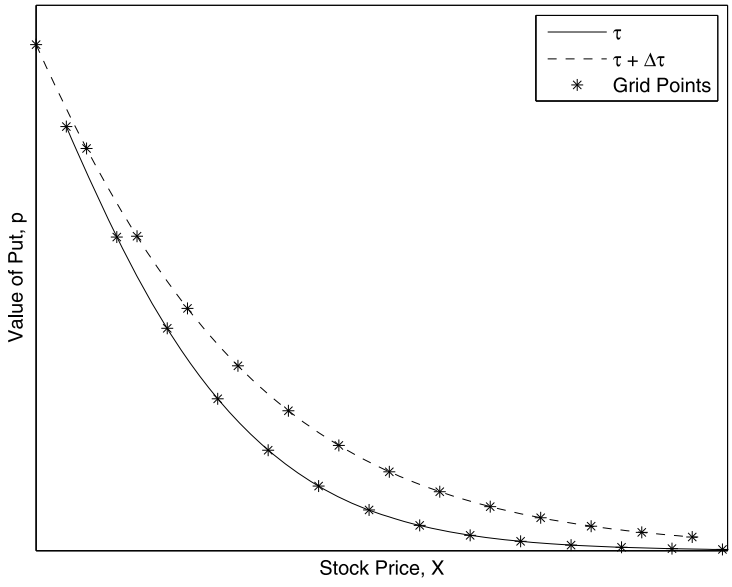

FIGURE 2.5. Illustrating different grid points used at two different times.

2.4. Modified Integral Method

In the previous sections we needed to evolve the boundary and the value function simultaneously because the boundary evolution equation requires a mixed derivative of the value function evaluated at the boundary. In this section, we present a numerical method that does not require the value function to be explicitly evolved with the boundary. This is achieved by using the integral representation of the American put option, as in Kim (1990). Although we do not directly extend this to American options with stochastic volatility, this method gives a good example of how to use boundary evolution equations to improve other numerical methods besides PDE methods.

The integral representation of the American put option states that the value of an American put is equal to the value of a European put, plus an early exercise premium. The early exercise premium is an integral of a function of the boundary. The value of an American put is

$$p(x,\tau) = f(x,\tau) + \int_0^\tau \left[rqe^{-r(\tau-u)}N(-d_2^*) - (r-b)xe^{-(r-b)(\tau-u)}N(-d_1^*)\right] \mathrm{d}u, \tag{2.14}$$

where

$$d_1^* = \frac{\log(x/c(u)) + \left(b + \frac{1}{2}\sigma^2\right)(\tau - u)}{\sigma\sqrt{\tau - u}} \quad \text{and} \quad d_2^* = d_1^* - \sigma\sqrt{\tau - u}.$$

Here $f(x, \tau)$ is the value of a European put, N is the standard normal cumulative distribution function, and we see that d_1^* and d_2^* are functions of the boundary.

Using this representation, if we know the value of the boundary between $\tau = 0$ and some time to expiry, τ_1, then we would like to express the value of the mixed derivative at τ_1 as some integral of the known boundary, which we can approximate using numerical integration. If this is possible we can then calculate the value of the boundary at

$\tau_1 + \Delta\tau$ using equation (2.6), and an ODE solver. This is exactly what we will do, however there are a few subtleties that arise that make this process complicated so we describe the process in four steps below.

Step 1: Initialization. To start this algorithm we must know the value of the boundary at a small time to expiry, as we did in the previous sections, because to calculate the mixed derivative using equation (2.14) we need something to integrate. To initialize this method we tried two different approaches. First we tried the asymptotic expansions for small τ, found in several papers mentioned in Section 1.1. We also tried the method we have used in previous sections: find the intersection of the European option with the intrinsic value of the put. It is somewhat surprising, but in numerical experiments we find that the method of finding the intersection is about five times more accurate than the asymptotic expansions that we tried, so we initialize with the binary search method. Once we know the boundary, we do not need to calculate the price function as we did in the previous sections because we will evaluate the mixed derivative as an integral of the boundary.

Step 2: Calculate $\frac{\partial^2 p}{\partial x \partial \tau}$. To calculate the mixed derivative we differentiate equation (2.14) first with respect to x and then τ. The derivative with respect to x is

$$\frac{\partial p}{\partial x} = \frac{\partial f}{\partial x} - \int_0^\tau \left[\frac{rq}{x\sigma\sqrt{\tau - u}} e^{-r(\tau-u)} N'(d_2^*) - \frac{r-b}{\sigma\sqrt{\tau-u}} e^{-(r-b)(\tau-u)} N'(d_1^*) + (r-b) e^{-(r-b)(\tau-u)} N(-d_1^*) \right] \mathrm{d}u. \tag{2.15}$$

When we differentiate equation (2.15) with respect to τ, using the Liebniz rule, and evaluate at $x = c(\tau)$ we find that $\frac{\partial^2 p}{\partial x \partial \tau} = \frac{\partial^2 f}{\partial x \partial \tau} + \infty - \infty$.

This does not mean that the derivative does not exist, rather it means that there is no analytical expression for the derivative, because there is no analytical expression for the integral in equation (2.15). This happens because the integrand in equation (2.15) blows up when $u = \tau$, even though it is still integrable.

To overcome this problem we employ numerical integration to calculate $\frac{\partial p}{\partial x}$, using equation (2.15), evaluated at $(c(\tau - \Delta\tau), \tau)$ and assume that $\lim_{u \to \tau} d_{1,2}^* = 0$. Then we can approximate the mixed derivative as

$$\left.\frac{\partial^2 p}{\partial x \partial \tau}\right|_{(c(\tau),\tau)} \approx \left(\left.\frac{\partial p}{\partial x}\right|_{(c(\tau-\Delta\tau),\tau)} + 1 \right) \Big/ \Delta\tau.$$

Here the "plus one" comes from the assumption that $\frac{\partial p}{\partial x}|_{(c(\tau-\Delta\tau),\tau-\Delta\tau)} = -1$.

The approximation of the integral in equation (2.15) is also not entirely straightforward because the integrand blows up when $u = \tau$, so any standard numerical approximation will undervalue the integral. To fix this we split the integral into two parts, first the integral from 0 to $\tau - \Delta\tau$ and then the integral from $\tau - \Delta\tau$ to τ. The first part of the integral can easily be approximated using any numerical integration technique and the second part of the integral can be approximated in closed form if we assume that $d_{1,2}^* = 0$ on the interval $[\tau - \Delta\tau, \tau]$. In fact, $d_{1,2}^* = 0$ when $u = \tau$ only if we evaluate at $x = c(\tau)$. Since we are trying to approximate the derivative at a value close to $c(\tau)$ we use $d_{1,2}^* = 0$ as an approximation. If $d_{1,2}^* = 0$ on the interval then $N'(d_{1,2}^*) = \frac{1}{\sqrt{2\pi}}$ and $N(d_{1,2}^*) = \frac{1}{2}$. This together with the fact that $\int \frac{e^{-\beta(\tau-u)}}{\sqrt{\tau-u}} \mathrm{d}u = -\sqrt{\frac{\pi}{\beta}} \mathrm{erf}(\sqrt{\beta(\tau-u)})$, where

erf is the error function, we can approximate the second part of the integral, and the mixed derivative quite accurately.

Step 3: Evolve $c(\tau)$. After the mixed derivative is calculated we hold it constant for one step and evolve the boundary one step using equation (2.6). Since we hold the mixed derivative constant for one step we see that equation (2.6) becomes an ODE and we can evolve it using any ODE solver. In numerical examples we use the explicit RK2 method. Once we evolve the boundary one step we repeat Steps 2 and 3 until we know the boundary at the desired time to expiration.

Step 4: Calculate the price of the put. Once we know the boundary for all values between 0 and τ we can use equation (2.14) to find the value of an American put at any value of x. Again we need to use numerical integration but this time the integral is very simple because the integrand does not blow up, so we can use any standard numerical integration technique. Numerical results are presented in the next section.

2.5. Numerical Results

To compare the speed and accuracy we compute the (long-dated) option values over the set of parameters presented on Table 3a in AitSahlia and Carr (1997). Here we assume that the underlying asset is a constant dividend paying stock and thus $b = r - \delta$, where δ is the dividend yield and we use $\hat{x} = 6.5q$. The value of the put is calculated when $x = 80$, 85, 90, 95, 100, 105, 110, 115, 120 for the parameter values $q = 100$, $\tau = 3$, $\sigma = 0.4$, $r = 0.06$, and $\delta = 0.02$. Then holding all other parameters fixed at this level we evaluate the at-the-money put with the parameters $r = 0.02$, 0.04, 0.08, 0.1, $\delta = 0$, 0.04, $\sigma = 0.3$, 0.35, 0.45, 0.5, and $\tau = 0.5$, 1, 1.5, 2, 2.5, 3.5, 4, 4.5, 5, 5.5. This leads to 21 sets of parameters where we evaluate the American put.

We compare these values to the values calculated using the very accurate, yet very slow, binomial tree method. We then compare this accuracy measure to the accuracy of four other computational methods: the finite difference moving boundary method in Muthuraman (2008), the Brennan–Schwartz method, the front fixing method in Nielsen et al. (2002) and the standard integral method in Carr et al. (1992). A more comprehensive comparison of other numerical methods can be found in Muthuraman (2008). The Brennan–Schwartz and the moving boundary method have some similarity to the static grid and dynamic grid methods since they find the boundary and evolve the price by time stepping. However in these methods the boundary is always considered to be at a grid point and the way it is found, by evolving equation (2.1) over a large domain several times, is much slower than our method, evolving an ODE. We compare to the standard integral method because we have created a modified integral method that uses the boundary evolution equation and we would like to see if this is advantageous. The front fixing method is considered too, because it also removes the moving boundary by a change of variables similar to the one considered here, however this method is very slow and inaccurate because it must solve a large system of nonlinear equations at each time step.

The measure of accuracy here is the same as the one used by Broadie and Detemple (1997), root mean squared relative error, RMSE, and we consider the "exact" price to be the average of a 10,000 and a 10,001 step binomial tree approximation, as in AitSahlia and Carr (1997). RMSE is defined as $\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\frac{\text{approx}_i - \text{exact}_i}{\text{exact}_i})^2}$ where the sum

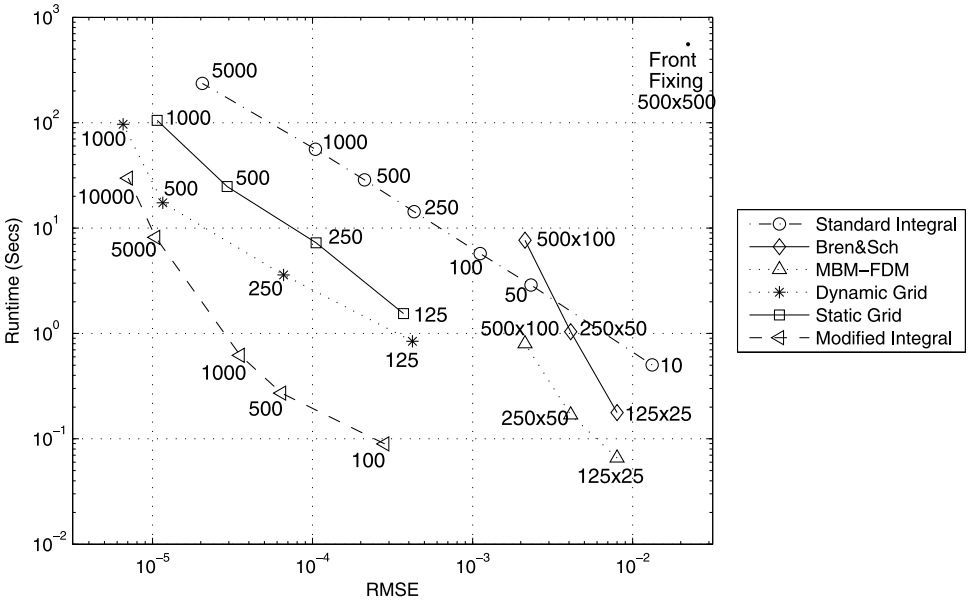

FIGURE 2.6. RMSE vs. runtime for constant volatility.

is taken over all numerical experiments, approx_i is the value of the ith put found by the approximate numerical method, and exact_i is the "exact" value of the put.

The measure of speed is simply average total computational time. We calculate the speed and error of these methods over several grid sizes and show the results in Figure 2.6. For the dynamic and static grid methods the labels refer to the number of spacial grid points; the number of grid points in time to expiry is determined by the CFL condition: the step size in τ is proportional to the square of the step size in x, which guarantees that the matrices used for evolution are positive definite. It is important that the evolution matrices be positive definite because if values on the main diagonal are negative then roundoff error can accumulate quickly, see Courant, Friedrichs, and Lewy (1967). For the modified and standard integral methods the labels refer to the number of time grid points, and for the Brennan–Schwartz, moving boundary and front fixing methods the labels refer to the number of space and time grid points. All computations were performed in Matlab on a PC with a 3.06 GHz processor and 4GB of RAM running Ubuntu Linux 10.10.

All analysis here was performed with $0 \leq \delta \leq r$. Unfortunately, when $r < \delta$ it can happen that our initial approximation of $c(\tau_o)$ is greater than $\frac{r}{\delta}q$. This means that the denominator of equation (2.6) is negative and the whole equation is positive, indicating that the boundary is increasing in time-to-expiry, which is clearly incorrect. To overcome this problem we can use other methods to initialize the American put, such as a few steps in the integral method or any variety of short-time asymptotic approximations. It seems that using a few steps in the integral method is favorable to short-time asymptotic approximations because the time required to initialize with the integral method does not increase total computational time by much and it typically results in less error than short-time asymptotic approximations. Alternatively if $\delta < 0$, (i.e., $b > r$), there is no change to the method and speed and accuracy are comparable to the existing results.

We can see that the static and dynamic grid methods perform better than the standard integral method in both computational time and accuracy. They also provide better

accuracy than the Brennan and Schwartz method and the moving boundary method. We also see that the front fixing method is the worst method considered, as is also seen in Muthuraman (2008).

The dynamic grid method is faster than the static grid method despite requiring more time steps because one time step of an explicit method, used in the dynamic grid method, can be faster than one time step of an implicit method, used in a static grid method. The dynamic grid method forces us to use an explicit method because of the nonlinearities. Each step in this explicit method results in a few matrix multiplications (depending on the order of the Runge–Kutta method) whereas the Crank–Nicolson method requires matrix multiplication and factorization, to solve a system of equations, at each time step. It is not possible to prestore the matrix factorization before evolving the system because at each step the matrix changes and thus the factorization changes as well. Even though the dynamic grid method takes more time steps than the static grid method, each step in the dynamic grid method is faster than a step in the static grid method and this trade off comes out in favor of the dynamic grid method for most mesh sizes.

Even more impressive than the static and dynamic grid methods is the modified integral method. The modified integral method offers a huge improvement over the standard integral method in both computational time and accuracy. It also out performs the static and dynamic grid methods, especially on a coarse grid. We do not directly extend the modified integral method to stochastic volatility but this would be an interesting direction for future research following Detemple and Tian (2002).

We would also like to know how error in these methods depends on grid size. To do this we will perform two convergence studies where we systematically decrease the step size in the x and τ variables. The first study will be performed on the Fixed Grid and Dynamic Grid methods. In this study when we reduce the step size in x (or ω) linearly, we reduce the step size in τ quadratically, to maintain the CFL condition. We then calculate the L^2 error and find the slope of the log step size versus the log error, this gives us the order of accuracy of the method.

To approximate the L^2 error we compute the price of the put for $x \in [80, 120]$ at $\tau = 3$ for the first parameter set described above. When we perform this regression we find the slope is 1.985 for the dynamic grid method and 1.419 for the fixed grid method, suggesting that the dynamic grid method is second-order accurate. The most likely reason that the fixed grid method loses some accuracy is the nonuniform grid spacing at the boundary; the small distance between the boundary and the first grid point can dominate finite difference calculations.

In the second study we examine the effect of $\Delta\tau$ on error. We perform this test only on the modified integral method. Here we systematically decrease the step size in τ, there is no step size in x, and again approximate the L^2 error over the same domain as in the previous example. When we perform this regression we find the slope is 0.949. This method is only first-order accurate, despite using what seemed to be a second-order finite difference method for $c(\tau)$, because the calculation of the mixed derivative at the boundary, which is only first-order accurate in time, dominates the error.

3. STOCHASTIC VOLATILITY

In this section, we seek the boundary evolution equation that characterizes the early exercise boundary when the dynamics of the underlying asset are modeled by a stochastic volatility process. We will also leverage on the derived equation to create a fast and

accurate numerical method to approximate the price of an American option. This time however we will only be able to implement the numerical method on a dynamic grid due to grid effects that will be explained later.

Working with stochastic volatility makes pricing options challenging since there are two space dimensions and one time dimension. The space dimensions are x, which represents the price of the underlying asset, and y, which represents the volatility, or some function of the volatility, of the underlying asset.

Unlike the constant volatility case, when we consider stochastic volatility there are several models in literature for the underlying dynamics of the asset's volatility. The popular models are the Heston (1993) model, the Hull and White (1987) model, the Scott (1987) model, and the Stein and Stein (1991) model. As each model uses a different stochastic process for volatility the PDE describing the risk neutral expectation is different for each model. In each of these the authors have worked on pricing European-style derivatives. Of the above models, the Heston model is the most popular and in the next sections we focus mostly on this model but also present some results for the other models above.

3.1. The Boundary Equation

As in the constant volatility case we must partition the computational domain into two distinct regions separated by the early exercise boundary. Now, however, the early exercise boundary is not just a function of time, but also the volatility level because different levels of volatility will lead to different optimal exercise policies. Before we can derive a PDE for the early exercise boundary we must first understand stochastic volatility models. We will begin working with a set of stochastic differential equations that are sufficiently general to accommodate the popular stochastic volatility models. The SDEs are

$$\begin{aligned} dX_t &= \mu X_t dt + f(Y_t) X_t dW_1, \\ dY_t &= \eta(Y_t) dt + \lambda(Y_t) dW_2, \\ \langle dW_1, dW_2 \rangle &= \rho dt. \end{aligned}$$

Here X_t is the stochastic process representing the price of the underlying asset, Y_t represents the volatility of the underlying asset, f, η, and λ are model specific functions, and ρ is the correlation between the two Brownian motions, W_1 and W_2. With these SDE's we can use a dynamic programming argument with Itô calculus and the no-arbitrage argument to write a PDE and boundary conditions that the value of the American put must satisfy in the nonexercise region of the domain, $\{(x, y, \tau)\colon \forall y, \tau \geq 0, x > c(y, \tau)\}$. Here we do not consider dividends for simplicity. The differential equation is

$$\frac{\partial p}{\partial \tau} = \frac{1}{2} x^2 f(y)^2 \frac{\partial^2 p}{\partial x^2} + \frac{1}{2} \lambda(y)^2 \frac{\partial^2 p}{\partial y^2} + \rho \lambda(y) f(y) x \frac{\partial^2 p}{\partial x \partial y} + r x \frac{\partial p}{\partial x} + \eta(y) \frac{\partial p}{\partial y} - rp, \tag{3.1}$$

with boundary conditions

$$p(c(y, \tau), y, \tau) = q - c(y, \tau), \tag{3.2}$$

$$\frac{\partial}{\partial x} p(c(y, \tau), y, \tau) = -1, \tag{3.3}$$

$$\frac{\partial}{\partial y} p(c(y,\tau), y, \tau) = 0, \quad \text{and} \tag{3.4}$$
$$p(x, y, 0) = \max(q - x, 0).$$

We also assume

$$\lim_{y\to\infty} \frac{\partial p}{\partial y} = 0, \tag{3.5}$$

which implies that $\lim_{y\to\infty} \frac{\partial c}{\partial y} = 0$. And for large x we use the same boundary condition as equation (2.5). Equations (3.3) and (3.4) are the smooth pasting conditions for stochastic volatility, as found in Fouque, Papanicolaou, and Sircar (2000). Now that we have the boundary conditions we next seek a differential equation that governs $c(y, \tau)$. We give the proof for the general formulation and later present the results for several specific models.

THEOREM 3.1. *If* $\frac{\partial c}{\partial \tau} \neq 0$ *and* $c(y, \tau)$ *is sufficiently smooth, the differential equation that governs* $c(y, \tau)$ *is*

$$\frac{\partial c}{\partial \tau} = -\frac{\partial^2 p(c, y, \tau)}{\partial x \partial \tau} \frac{1}{2rq} \left(f(y)^2 c^2 - 2\rho\lambda(y) f(y) c \frac{\partial c}{\partial y} + \lambda(y)^2 \left(\frac{\partial c}{\partial y} \right)^2 \right). \tag{3.6}$$

Proof. Similar to the derivation of equation (2.7), in stochastic volatility we also have

$$\frac{\partial}{\partial \tau} p(c(y,\tau), y, \tau) = 0. \tag{3.7}$$

Next we differentiate (3.3) with respect to y and τ, and (3.4) with respect to y giving us

$$\frac{\partial^2 p}{\partial x^2} \frac{\partial c}{\partial y} + \frac{\partial^2 p}{\partial x \partial y} = 0, \tag{3.8}$$

$$\frac{\partial^2 p}{\partial x^2} \frac{\partial c}{\partial \tau} + \frac{\partial^2 p}{\partial x \partial \tau} = 0, \tag{3.9}$$

and

$$\frac{\partial^2 p}{\partial x \partial y} \frac{\partial c}{\partial y} + \frac{\partial^2 p}{\partial y^2} = 0. \tag{3.10}$$

Now combine (3.8) and (3.10) to see that

$$\frac{\partial^2 p}{\partial y^2} = \frac{\partial^2 p}{\partial x^2} \left(\frac{\partial c}{\partial y} \right)^2. \tag{3.11}$$

Next evaluate (3.1) at the boundary and substitute in (3.2), (3.3), (3.4), (3.8), and (3.11), which gives us

$$0 = \frac{1}{2} f(y)^2 c^2 \frac{\partial^2 p}{\partial x^2} - \rho\lambda(y) f(y) c \frac{\partial^2 p}{\partial x^2} \frac{\partial c}{\partial y} + \frac{1}{2} \lambda(y)^2 \left(\frac{\partial c}{\partial y} \right)^2 \frac{\partial^2 p}{\partial x^2} - rq. \tag{3.12}$$

Finally plug in (3.9) to equation (3.12) and rearrange terms to obtain the desired result, (3.6). □

Now that we have the formula for the general stochastic volatility formulation we plug in the model specific functions, f, η, and λ, and describe the boundary equations for the four models above in Table 3.1. In all of these models the market price of risk is assumed to be zero but it could be inserted into the differential equations without much effort because the coefficient of the first derivative in y, which is where the market price of risk enters the system, is not present in the boundary evolution equation.

In the statement of Theorem 3.1 we only derive the boundary evolution equation when $\frac{\partial c}{\partial \tau} \neq 0$ for all values of y. This guarantees that we do not divide by zero when plugging equation (3.9) into (3.12). If this is not true, then the boundary just does not move at that point. It seems however for the Heston model and the Hull and White model that as $y \to 0$ we also have $\frac{\partial c}{\partial \tau} \to 0$ for all values of τ. This would mean that $c(0, \tau) = q$ and $p(x, 0, \tau) = 0$ for all τ and $x \geq q$. For the Hull and White model this is not surprising because the variance in this model follows a Geometric Brownian Motion, which stays at zero forever if the process is ever zero, almost surely. This means that the value of the underlying becomes deterministic and thus an out-of-the-money put can have no value when $y = 0$, which can be used as a boundary condition for the Hull and White model.

This point is subtle because even though a Geometric Brownian Motion can never reach zero, if it starts at a positive value, the PDE for the value function needs a boundary condition. The boundary condition chosen here needs to agree with the dynamics of the stochastic process, and since a Geometric Brownian Motion that starts at zero must stay at zero, this is the boundary condition that we must use.

The above economic reasoning, however, does not make sense for the Heston model because the variance follows a square root process which becomes positive immediately after hitting zero, almost surely (for certain parameter values satisfying the Feller Condition, zero is inaccessible to the variance process, like Geometric Brownian Motion, but we still need a boundary condition.) This means that the value of the underlying cannot be deterministic and thus an out-of-the-money put must have positive value when $y = 0$, implying that $\lim_{y \to 0^+} \frac{\partial c}{\partial y} = -\infty$.

In this case the rate that the derivative explodes must be very specific. It must go to infinity like $\frac{-1}{\sqrt{y}}$. If it goes to infinity any faster then the last term in the Heston boundary equation will go to infinity and so will the whole boundary equation. If it goes to infinity any slower then the last term will go to zero and so will the whole boundary equation. If the derivative does go to infinity at the right speed then the last term becomes indeterminate, which makes

$$\frac{\partial c(0, \tau)}{\partial \tau} = -\frac{\partial^2 p(c, y, \tau)}{\partial x \partial \tau} \frac{v^2}{2rq} \gamma. \tag{3.13}$$

Although the above constant, γ, is unknown we can interpret this as a boundary condition for the Heston model, which will be explained in further detail in the next section.

Figure 3.1 shows the state space partitioned to the exercise region and the continuation region for the Heston model. The two regions are separated by the early exercise surface. The exercise region is below the surface and the price of the put is equal to its intrinsic value there. The continuation region is above the surface and there the price of the put is governed by equation (3.1).

TABLE 3.1.
Boundary Evolution Equations for Several Popular Stochastic Volatility Models

Model	Stochastic Process	Boundary Equation
Heston	$dX_t = \mu X_t dt + \sqrt{Y_t} X_t d W_1$	$\frac{\partial c}{\partial \tau} = -\frac{\partial^2 p(c,y,\tau)}{\partial x \partial \tau} \frac{1}{2rq}(yc^2 - 2\rho v y c \frac{\partial c}{\partial y} + v^2 y (\frac{\partial c}{\partial y})^2)$
$\rho \neq 0$	$dY_t = \kappa(m' - Y_t)dt + v\sqrt{Y_t} d W_2$	
Hull & White	$dX_t = \mu X_t dt + \sqrt{Y_t} X_t d W_1$	$\frac{\partial c}{\partial \tau} = -\frac{\partial^2 p(c,y,\tau)}{\partial x \partial \tau} \frac{1}{2rq}(yc^2 + \alpha_2^2 y^2 (\frac{\partial c}{\partial y})^2)$
$\rho = 0$	$dY_t = \alpha_1 Y_t dt + \alpha_2 Y_t dW_2$	
Scott	$dX_t = \mu X_t dt + e^{Y_t} X_t d W_1$	$\frac{\partial c}{\partial \tau} = -\frac{\partial^2 p(c,y,\tau)}{\partial x \partial \tau} \frac{1}{2rq}(e^{2y} c^2 + \beta^2 (\frac{\partial c}{\partial y})^2)$
$\rho = 0$	$dY_t = \alpha(m - Y_t)dt + \beta dW_2$	
Stein & Stein	$dX_t = \mu X_t dt + \lvert Y_t \rvert X_t dW_1$	$\frac{\partial c}{\partial \tau} = -\frac{\partial^2 p(c,y,\tau)}{\partial x \partial \tau} \frac{1}{2rq}(y^2 c^2 + \beta^2 (\frac{\partial c}{\partial y})^2)$
$\rho = 0$	$dY_t = \alpha(m - Y_t)dt + \beta dW_2$	

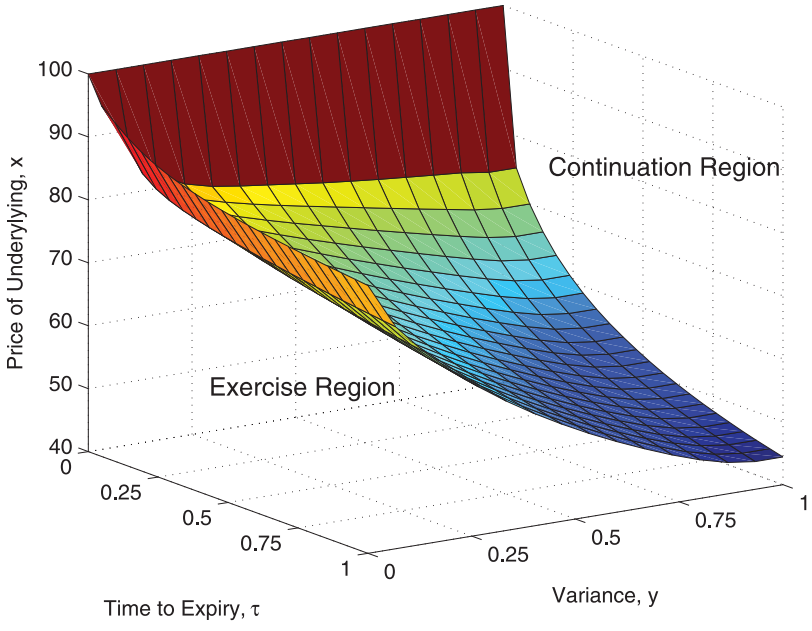

FIGURE 3.1. Partitioned state space for Heston model.

3.2. Numerical Method on a Dynamic Grid

In this section we will focus only on the Heston model of stochastic volatility. We want to transform the no exercise region to a simpler domain that allows for standard finite difference methods. We transform the domain $\{(x, y, \tau)\colon \forall y, \tau \geq 0, x > c(y, \tau)\}$ to $\mathbb{R}^3_+$. The change of variable we use to transform our domain is

$$\begin{aligned} \omega &= x - c(y, \tau) \\ g(\omega, y, \tau) &= p(x, y, \tau). \end{aligned} \tag{3.14}$$

This change of variable involves derivatives in the y variable. The second derivative and the mixed derivative lead to several nonlinear terms in the resulting PDE. Equation (3.1) for the Heston model is

$$\frac{\partial p}{\partial \tau} = \frac{1}{2} y x^2 \frac{\partial^2 p}{\partial x^2} + \frac{1}{2} v^2 y \frac{\partial^2 p}{\partial y^2} + \rho v y x \frac{\partial^2 p}{\partial x \partial y} + r x \frac{\partial p}{\partial x} + \kappa(m' - y)\frac{\partial p}{\partial y} - rp. \tag{3.15}$$

We use the chain rule to find the pricing equation for the g function and the corresponding boundary equation, which are

$$\begin{aligned} \frac{\partial g}{\partial \tau} &= \frac{1}{2} y(\omega + c(y, \tau))^2 \frac{\partial^2 g}{\partial \omega^2} + \frac{1}{2} v^2 y \left(\frac{\partial^2 g}{\partial \omega^2} \left(\frac{\partial c}{\partial y} \right)^2 - \frac{\partial g}{\partial \omega} \frac{\partial^2 c}{\partial y^2} - 2 \frac{\partial c}{\partial y} \frac{\partial^2 g}{\partial \omega \partial y} + \frac{\partial^2 g}{\partial y^2} \right) \\ &\quad + \rho v y(\omega + c(y, \tau)) \left(\frac{\partial^2 g}{\partial \omega \partial y} - \frac{\partial c}{\partial y} \frac{\partial^2 g}{\partial \omega^2} \right) + r(\omega + c(y, \tau)) \frac{\partial g}{\partial \omega} + \kappa(m' - y) \\ &\quad \times \left(\frac{\partial g}{\partial y} - \frac{\partial g}{\partial \omega} \frac{\partial c}{\partial y} \right) - rg + \frac{\partial g}{\partial \omega} \frac{\partial c}{\partial \tau} \end{aligned} \tag{3.16}$$

and

$$\frac{\partial c}{\partial \tau} = -\frac{\partial^2 g(0, y, \tau)}{\partial \omega \partial \tau} \frac{1}{2rq} \left(yc^2 - 2\rho v y c \frac{\partial c}{\partial y} + v^2 y \left(\frac{\partial c}{\partial y} \right)^2 \right). \tag{3.17}$$

Given this change of variables we seek a numerical method that exploits the equations for boundary and price evolution. The method presented here can be summarized in a three-step process.

Step 1: Initialization. Similar to the constant volatility case, we cannot start the numerical method with the initial conditions when $\tau = 0$, and as such we need to approximate the value of the American option a short time before expiry as a European option. There are two ways to approximate the value of the European option. First we could use the semi closed form solution to European options under the Heston model to find the price of the European a short time before expiry, the details of which can be found in Heston (1993) or Gatheral (2006). Alternatively we could use the constant volatility Black–Scholes equation to find the value of the put a short time before expiration. It might seem that this simplistic method would lead to large error, but it turns out that the two methods have comparable accuracy and the second is significantly faster than the first. The reason is that the functions being integrated in the solution to the European put under the Heston model are highly oscillatory and are dampened very slowly for small values of τ. This makes approximating this integral a very slow process because a large integration domain is required with a fine integration mesh, and so for numerical tests we simply use the Black–Scholes equation to initialize p.

To initialize we need to divide the y domain uniformly between 0 and $\hat{y}$, where $\hat{y}$ is the maximal value of the computational domain. Here again the value of $\hat{y}$ needs to be large enough so that the boundary condition in Equation (3.5) is approximately true for all values of x. At each grid point in y we perform a binary search to find the intersection of the value of the European option and the intrinsic value of the option as in Figure 2.2. If we use the Black–Scholes formula to get the value of the European then we need to set the variance equal to the y grid value. We initialize the boundary at each y grid point as the location of the intersection. Then for each value of y we find the value of the European at n equally spaced grid points, in x, larger than the boundary, where n is chosen large enough so that the boundary condition (2.5) is satisfied. We find that the initial boundary is deceasing in y and as such the maximal value of x for each value of y is also decreasing. After we find the price of the European at all of these grid points we transform the domain using equation (3.14). Figure 3.2 shows how the computational domain looks before the transformation.

Step 2: Calculate $\frac{\partial^2 g(0, y, \tau)}{\partial \omega \partial \tau}$. As in the constant volatility case the hardest part of the algorithm is finding the mixed derivative at the boundary. In the constant volatility case we discussed two ways to calculate this derivative, one in the fixed grid method and one in the dynamic grid method. Here since we only work on a dynamic grid we simply evolve a few grid points greater than the boundary for every value of y according to equation (3.16) without the last term, the grid speed term, using an explicit Runge–Kutta method. We use a standard one-sided finite difference method to calculate the value of the x derivative at the boundary for every value of y after this partial evolution. Then using this value with equation (3.3) we can approximate the value of the mixed derivative by using a first-order finite difference method.

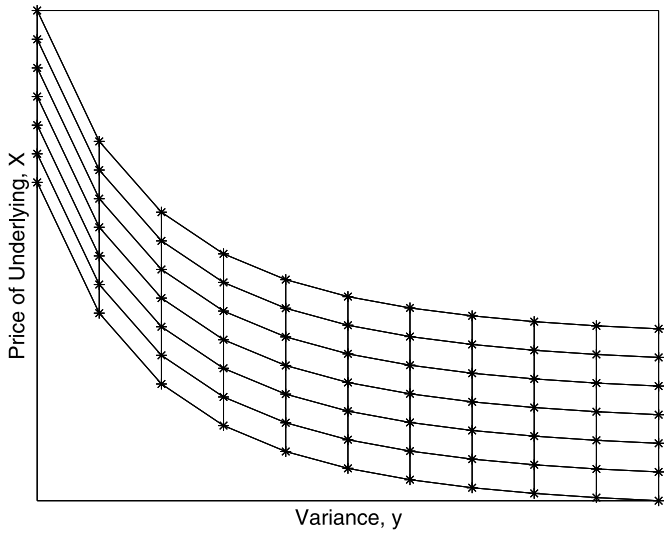

FIGURE 3.2. Computational grid before transformation.

Step 3: Evolve c and g in time to expiry. As opposed to the method used for constant volatility, we linearize equation (3.16) so that we can use an implicit method to step backwards in time, which dramatically reduces the number of steps required in time to expiry when compared to an explicit method. In constant volatility we could have also linearized the price evolution equation in the dynamic grid section to use an implicit method. However on a fine grid linearization accounts for a large portion of the numerical error and so we only use an explicit method. In stochastic volatility it is not practical to use a fine grid because there are two space dimensions which greatly increases the total number of grid points and therefore we linearize equation (3.16).

We see that in equation (3.16) all the nonlinearities come from multiplying derivatives of g with derivatives of c. This means that if we can approximate the derivatives of c then we can use them to linearize the evolution equation for g. To linearize this equation we must get the first- and second-order derivatives of the boundary with respect to y. To do this we simply evolve the boundary one step using a Runge–Kutta method and compute the derivatives for the boundary at the τ and $(\tau+1)^{\text{st}}$ steps using standard finite difference methods. Then using the values computed here we plug them into Crank–Nicolson matrices A and B, where A and B are block tridiagonal matrices satisfying the equation $A \cdot g^{\tau} = B \cdot g^{\tau+1}$. Here we plug the values of the derivatives before the evolution into the A matrix and the values of the derivatives after the evolution into the B matrix. One important fact to remember is that at each step the matrices A and B must be recalculated because the boundary and the derivatives of the boundary have changed. After we evolve the price function we let the boundary be equal to the value at the $(\tau+1)^{\text{st}}$ step.

There is still a boundary condition we need to address: the boundary when y goes to 0. For this we simply assume that the constant in equation (3.13) is attained at the second y grid point and that the value of p evolves with the standard PDE when 0 is inserted for y, which eliminates several terms. After we evolve g one step we repeat Steps 2 and 3 until we reach the desired time to expiration and change the variables back to x and p.

We cannot adopt this method onto a fixed grid, as we did with constant volatility, because the boundary is decreasing in y. Say for a specific y value the boundary is between

the 99th and 100th x grid values and for the next y value the boundary is between the 98th and 99th x grid values. We will then get a discontinuity in the calculation of the mixed derivative when this happens. This discontinuity in the mixed derivative leads to a discontinuous boundary which in turn leads to large error in the price of the put and as such we need a dynamical grid method. The boundary is also decreasing in time to expiry and so this phenomenon could also occur in the τ variable. The effect, however, is less drastic in τ than in y because at each discrete step in τ we numerically approximate derivatives in y, whereas we relate derivatives in τ to derivatives in y using equation (3.16) removing the need for continuous derivatives in τ. This relationship is why we were able to use a static grid for constant volatility.

Also, as in the constant volatility case we add extra grid points to each value of y every time that the boundary decreases below a certain value. This again has the benefit of maintaining accuracy for options that are out of the money.

3.3. Numerical Results

Numerical comparison of speed and accuracy is more challenging for stochastic volatility than for constant volatility because finding a "true" price for the option is not clear. In this section we only compute the price of the put for eight set of parameters, the "true" values were calculated by Jari Toivanen using his component-wise splitting method on a very fine mesh. The value of the put is calculated when $x = 8, 9, 10, 11, 12$, $y = 0.0625, 0.25$, and $\tau = 0.25$ for the parameter values $r = 0.1$, $v = 0.9$, $\kappa = 5$, $m' = 0.16$, and $\rho = 0.1$. Then holding all other parameters fixed we evaluate the puts with the parameters $r = 0.08, 0.12$, $v = 0.7, 1.1$, $\kappa = 2.5$, and $\rho = 0.05, 0.15$.

We compare our method to two existing methods, the PSOR and the moving boundary method, MBM, presented in Choklingham and Muthuraman (2011), in Figure 3.3. We only compare our method against these methods because although the PSOR method is quite slow, Ikonen and Toivanen (2008) find that it is the simplest to implement, and in Chockalingham and Muthuraman (2011) the authors find that the moving boundary method was the fastest method tested. As in the constant volatility case we plot the root mean squared relative error of the methods versus the total computational time. For the moving boundary method and the PSOR the labels refer to the number of x, y, and τ grid points. For the dynamic grid method the labels refer to the number of x and y grid points, and the number of time grid points is determined by the CFL condition.

The nonlinearities of the dynamic grid method unfortunately cause the necessary number of steps in time to expiry to be quite large, despite the linearization of equation (3.16), this however is offset by the speed with which each time step is executed versus the moving boundary method and the PSOR. Both of these methods must search for the early exercise boundary while our method knows exactly where it is. We see that for the coarsest grid the moving boundary method is slightly better than our method, however on finer grids our method performs significantly better. For the finest grid our method is almost three times faster than the moving boundary method.

4. CONCLUDING REMARKS

Boundary evolution equations have significant computational benefit when one relies on dynamic grids that are evolved with the boundary during the solution process. The key insight into the construction of efficient numerical methods is that we do not have to iteratively guess the location of the boundary at each step, rather the boundary evolution

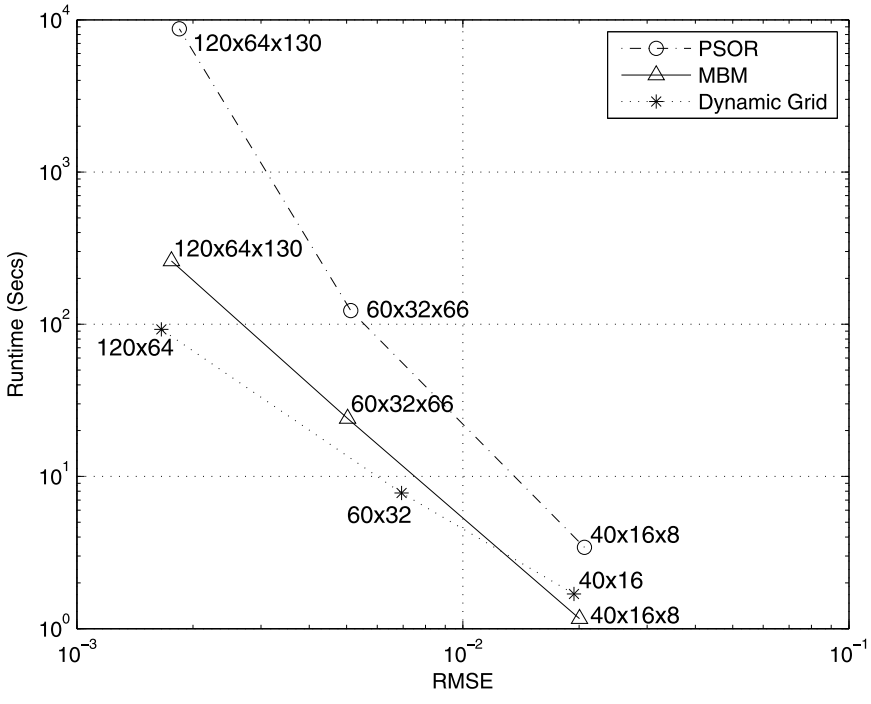

FIGURE 3.3. RMSE vs. runtime for stochastic volatility.

equation tells us its location. Moreover, by evolving the grid along with the boundary one gets the added benefit of minimizing the error in approximating the boundaries with a predefined grid structure.

The American option pricing problems studied here belong to the much larger class of optimal stopping problems in stochastic control. Most optimal stopping problems do not have analytical solutions and are difficult to solve, especially when the complexity of the state evolution equation increases. In many cases the location of the boundary that separates the stopping and continuation regions is of primary interest. As such boundary evolution equations can provide insight into the structure and nature of these boundaries. The derivation of the boundary evolution equations rely on the smooth pasting condition at the interface between the stopping and continuation regions. Similar smooth pasting conditions are also common in several derivative securities and other optimal stopping problems, such as simultaneous hypothesis testing and earliest detection problems. See Peskir and Shiryaev (2006) for examples of other optimal stopping problems.

In the Black–Scholes setting we presented a modified integral method for pricing American options that relied on an integral representation of the price of the American option. This method proved to be extremely fast and accurate in the simple case of Black–Scholes. An extension to multifactor models of the integral representation has been presented in Detemple and Tian (2002) and an interesting direction of future research would be to apply the boundary evolution equations for stochastic volatility found in this paper to a modified integral method for multifactor models.

Two other classes of stochastic control problems whose solutions are characterized by free-boundary problems are singular and impulse control. In these problems the state process is not terminated at the boundary, but a control is applied to it. Both deriving boundary evolution equations and constructing computational methods for these would be interesting future work. The ideas in this paper cannot be immediately extended to

optimal stopping problems with multiple boundaries and this would be interesting future work as well.

REFERENCES

AitSahlia, F., and P. Carr (1997): American Options: A Comparison of Numerical Methods, in *Numerical Methods in Finance*, L. Rogers and D. Talay, eds. Cambridge, UK: Cambridge University Press, pp. 67–87.

Black, F., and M. Scholes (1973): The Pricing of Options and Corporate Liabilities, *J. Polit. Econ.* 81, 637–654.

Brennan, M., and E. Schwartz (1977): The Valuation of American Put Options, *J. Finance* 32, 449–642.

Broadie, M., and J. Detemple (1996): American Option Valuation: New Bounds, Approximations, and a Comparison of Existing Methods, *Rev. Finan. Stud.* 9, 1211–1250.

Broadie, M., and J. Detemple (1997): Recent Advances in Numerical Methods for Pricing Derivative Securities, in *Numerical Methods in Finance*, L. Rogers and D. Talay, eds. Cambridge, UK: Cambridge University Press, pp. 43–66.

Broadie, M., J. Detemple, E. Ghysels, and O. Torres (2000): American Options with Stochastic Dividends and Volatility: A Nonparametric Investigation, *J. Econometr.* 94, 53–92.

Broadie, M., and P. Glasserman (1997): Pricing American-Style Securities by Simulation, *J. Econ. Dyn. Contl.* 21, 1323–1352.

Bunch, D., and H. Johnson (2002): The American Put Option and Its Critical Stock Price, *J. Finance* 55, 219–237.

Carr, P., R. Jarrow, and R. Myneni (1992): Alternative Characterizations of American Put Options, *Math. Finance* 2, 87–106.

Chen, S., B. Merriman, S. Osher, and P. Smereka (1997): A Simple Level Set Method for Solving Stefan Problems, *J. Comput. Phys.* 135, 8–29.

Chen, X., and J. Chadam (2007): A Mathematical Analysis of the Optimal Exercise Boundary for American Put Options, *SIAM J. Math. Anal.* 38, 1613–1641.

Chockalingam, A., and K. Muthuraman (2011): American Options under Stochastic Volatility, *Operat. Res.* 59, 793–804.

Clarke, N., and K. Parrott (1999): Multigrid for American Option Pricing with Stochastic Volatility, *Appl. Math. Finance* 6, 177–195.

Courant, R., K. Friedrichs, and H. Lewy (1967): On the Partial Difference Equations of Mathematical Physics, *IBM J. Res. Develop.* 11, 215–234.

Cox, J., S. Ross, and M. Rubinstein (1979): Option Pricing: A Simplified Approach, *J. Finan. Econ.* 7, 229–263.

Crank, J., and P. Nicolson (1947): A Practical Method for Numerical Evaluation of Solutions of Partial Differential Equations of the Heat-Conduction Type, *Proc. Camp. Phil. Soc.* 43, 50–67.

Detemple, J., and W. Tian (2002): The Valuation of American Options for a Class of Diffusion Processes, *Manage. Sci.* 48, 917–937.

Evans, J., R. Kuske, and J. Keller (2002): American Options on Assets with Dividends Near Expiry, *Math. Finance* 12, 219–237.

Fouque, J., G. Papanicolaou, and K. Sircar (2000): *Derivatives in Financial Markets with Stochastic Volatility*, Cambridge: Cambridge University Press.

Gatheral, J. (2006): *The Volatility Surface: A Practitioner's Guide*, Hoboken, NJ: Wiley.

Goodman, J., and D. Ostrov (2002): On the Early Exercise Boundary of the American Put Option, *SIAM J. Appl. Math.* 62, 1823–1835.

HAYES, E. (2006): The Application of a Semi-analytical Method for Computing Asymptotic Approximations to Option Prices, PhD thesis, Courant Institute of Mathematical Sciences, New York University.

HESTON, S. (1993): A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options, *Rev. Finan. Stud.* 6, 327–343.

HUANG, J., M. SUBRAHMANYAM, and G. YU (1996): Pricing and Hedging American Options: A Recursive Integration Method, *Rev. Finan. Stud.* 9, 277–300.

HULL, J., and A. WHITE (1987): The Pricing of Options on Assets with Stochastic Volatilities, *J. Finance* 42, 281–300.

IKONEN, S., and J. TOIVANEN (2007): Componentwise Splitting Methods for Pricing American Options Under Stochastic Volatility, *Int. J. Theor. Appl. Finance* 10, 331–361.

IKONEN, S., and J. TOIVANEN (2008): Efficient Numerical Methods for Pricing American Options under Stochastic Volatility, *Numer. Methods Partial Different. Equat.* 24, 104–126.

ISERLES, A. (2008): *A First Course in the Numerical Analysis of Differential Equations*, Cambridge, UK: Cambridge University Press.

JACKA, S. (1991): Optimal Stopping and the American Put, *Math. Finance* 1, 1–14.

KARATZAS, I., and S. SHREVE (1998): *Methods of Mathematical Finance*, New York: Springer.

KIM, I. (1990): The Analytic Valuation of American Options, *Rev. Finan. Stud.* 3, 547–772.

LAWRENCE, P., and S. SALSA (2009): Regularity of the Free Boundary of an American Option on Several Assets, *Comm. Pure Appl. Math.* 62, 969–994.

LONGSTAFF, F., and E. SCHWARTZ (2001): Valuing American Options by Simulation: Simple Least-Squares Approach, *Rev. Finan.Stud.* 14, 113–147.

MERTON, R. (1992): *Continuous-Time Finance*, Oxford: Blackwell.

MUTHURAMAN, K. (2008): A Moving Boundary Approach to American Option Pricing, *J. Econ. Dyn. Contl.* 32, 3520–3537.

MYNENI, R. (1992): The Pricing of the American Option, *Ann. Appl. Probab.* 2, 1–23.

NIELSEN, B., O. SKAVHAUG, and A. TVEITO (2002): Penalty and Front-Fixing Methods for the Numerical Solution of American Option Problems, *J. Comput. Finance* 5, 69–97.

PESKIR, G., and A. SHIRYAEV (2006): *Optimal Stopping and Free-Boundary Problems*, Basel: Birkhäuser Verlag.

SCOTT, L. (1987): Option Pricing When the Variance Changes Randomly: Theory, Estimation, and an Application, *J. Finan. Quantit. Anal.* 22, 419–438.

STAMICAR, R., D. SEVCOVIC, and J. CHADAM (1999): The Early Exercise Boundary for the American Put Near Expiry: Numerical Approximations, *Can. Appl. Math. Quart.* 7, 427–444.

STEIN, E., and J. STEIN (1991): Stock Price Distributions with Stochastic Volatility: An Analytic Approach, *Rev. Finan. Stud.* 4, 727–752.

THOMPSON, J., Z. WARSI, and C. MASTIN (1985): *Numerical Grid Generation: Foundations and Applications*, New York: Elsevier Science.

TILLEY, J. (1993): Valuing American Options in a Path Simulation Model, *Transactions of the Society of Actuaries* 45, 83–104.

VAN MOERBEKE, P. (1975): On Optimal Stopping and Free Boundary Problems, *Arch. Rat. Mech. Anal.* 60, 101–148.

WILMOTT, P. (1998): *Derivatives*, Chichester: Wiley.

WU, L., and Y. KWOK (1997): A Front-Fixing Finite Difference Method for the Valuation of American Options, *J. Finan. Eng.* 6, 83–97.