Foundations and Trends® in
Stochastic Systems
Vol. 1, No. 4 (2006) 259–341

DOI: 10.1561/0900000006

now
the essence of knowledge

Solving Free-boundary Problems with Applications in Finance

Kumar Muthuraman[1] and Sunil Kumar[2]

[1] *McCombs School of Business, University of Texas at Austin, 1 University Sta B6000, Austin, TX 78712, USA, kumar@mail.utexas.edu*
[2] *Graduate School of Business, Stanford University, 518 Memorial Way, Stanford, CA 94305, USA, skumar@stanford.edu*

Abstract

Stochastic control problems in which there are no bounds on the rate of control reduce to so-called free-boundary problems in partial differential equations (PDEs). In a free-boundary problem the solution of the PDE and the domain over which the PDE must be solved need to be determined simultaneously. Examples of such stochastic control problems are singular control, optimal stopping, and impulse control problems. Application areas of these problems are diverse and include finance, economics, queuing, healthcare, and public policy. In most cases, the free-boundary problem needs to be solved numerically.

In this survey, we present a recent computational method that solves these free-boundary problems. The method finds the free-boundary by solving a sequence of fixed-boundary problems. These fixed-boundary problems are relatively easy to solve numerically. We summarize and unify recent results on this *moving boundary* method, illustrating its

application on a set of classical problems, of increasing difficulty, in finance. This survey is intended for those are primarily interested in computing numerical solutions to these problems. To this end, we include actual Matlab code for one of the problems studied, namely, American option pricing.

1

Introduction

Singular stochastic control problems are those problems in which control effort can effect instantaneous displacement in state. A wide variety of problems can be modeled as singular stochastic control problems. A representative list of applications include economics [16], portfolio optimization in finance [15], dynamic control of queueing networks [22], revenue management [11], and environmental clean-up issues in public policy [28].

Despite their wide applicability, singular control problems are not analytically tractable except in very special cases. Therefore, one is forced to solve these problems numerically. Based on the application, various numerical methods have been proposed for solving such problems.

Our goal in this survey is to describe a general yet efficient numerical method for solving such problems. As a consequence we rely quite heavily on our own past work in this area, and borrow heavily from our papers. Our goal is to provide a unified treatment of this method emphasizing application in Finance.

Our method makes use of the special structure of these problems, namely, that optimal policies are characterized by so-called *regions of*

inaction. So we reduce the problem of finding the optimal policies to searching for the right region of inaction. This is still a difficult problem because there is no explicit characterization of the optimal region. Rather it is implicitly specified by the solution of a partial differential equation, whose domain it is. Thus, we are faced with a so-called *free-boundary problem* where the solution of a PDE and the domain of which it must be solved need to be simultaneously determined. Our method solves these free-boundary problems by reducing them into a sequence of fixed-boundary problems which are relatively easy to solve numerically. The key to our method is a boundary update procedure that allows us to construct the next fixed-boundary problem from the solution of the previous one.

A valuable by-product of our method is that it is capable of solving other stochastic control problems that can be cast as free-boundary problems but are not singular control problems *per se.* An important class of problems that fall in this category are optimal stopping problems. A very important example of an optimal stopping problem is the American option pricing problem, and our method is applicable to these problems as well, as we will discuss in a later section. In the same vein, our method is also applicable to impulse control problems [18], but we do not discuss these problems in this survey.

Rather than describe the method in vague words, we provide a simple, concrete illustration using a one-dimensional singular control problem. Restricting attention initially to one-dimensional problems allows us to achieve two objectives. We can provide a description of the procedure that is easier to comprehend, and we illustrate the kind of theoretical guarantees on the behavior of the procedure that can be obtained.

1.1 Motivation: Controlled Brownian Motion

Consider the problem of using two non-negative, nondecreasing, RCLL processes to control a given continuous path $w(\cdot)$ as

$$x(t) = x + \mu t + \sigma w(t) + L(t) - U(t), \quad t \geq 0, \tag{1.1}$$

where μ and $\sigma > 0$ are constants, and the initial state $x \in [0,1]$. Suppose that we needed $x(t) \in [0,1]$ for all t, and furthermore, we were

interested in maintaining $x(t)$ as close to $\bar{x}$ in the sense of minimizing $\int_0^\infty e^{-\lambda t} h(x(t))dt$, where $\lambda > 0$ is a discount rate, and $h(y) = (y - \bar{x})^2$, where $\bar{x} \in (0,1)$. For now, we impose no probabilistic structure on the problem. This problem is trivially solved. If $x \neq \bar{x}$ we simply make an initial jump using either L or R to move $x(0+)$ to $\bar{x}$ and maintain $x(t) = \bar{x}$ for all $t > 0$ using L and U. Note that we may have jumps in L or U even though w is continuous. In particular, there are no rate constraints on L and U. Hence the name *singular control*.

Now consider costs of control as well. Suppose, we have cost rates $c > 0$ and $r > 0$ such that the overall cost of using controls L and U is $\int_0^\infty e^{-\lambda t} h(x(t))dt + \int_0^\infty e^{-\lambda t} c dL(t) + \int_0^\infty e^{-\lambda t} r dU(t)$, where the last two terms are interpreted as Reimann–Steiljes integrals. Now the choice of controls is no longer obvious. Attempting to make $h(x(t))$ small comes at the price of incurring control costs. If the path w is not of bounded variation, then costs of control incurred by attempting to maintain $x(t) \equiv \bar{x}$ is prohibitive. So in order to trade-off the holding cost h against control costs, one pick controls that do nothing as long as $x(t)$ is close to $\bar{x}$ but intervene only when it has deviated sufficiently, i.e., it has hit the boundary of an interval around $\bar{x}$. It is not hard to see that the same interval cannot be appropriate for all paths. So we can no longer hope for path-wise solutions. To solve this problem we need to impose additional probabilistic structure on it.

Let $(\Omega, \mathcal{F}, P)$ be a probability space and $\{\mathcal{F}_t,\, t \in \mathbf{R}_+\}$ be a right continuous filtration on this space and let w be a standard ($\mathbf{R}$-valued) Brownian motion with respect to this filtration. Let L and U be RCLL, non-negative, nondecreasing and adapted to $\mathcal{F}_t$. Now suppose that we are again interested in the controlled process $x(\cdot)$ specified by (1.1), and in minimizing the expected infinite horizon discounted cost (also called a value function) among all admissible policies L and U.

$$J(x, L, U) = E_x \left[\int_0^\infty e^{-\lambda t} h(x(t))dt + \int_0^\infty e^{-\lambda t} c dL(t) + \int_0^\infty e^{-\lambda t} r dU(t) \right], \qquad (1.2)$$

For this problem to make sense we restrict attention to only those controls for which $E_x\big[\int_0^\infty e^{-\lambda t} dL(t)\big] < \infty$ and $E_x\big[\int_0^\infty e^{-\lambda t} dU(t)\big] < \infty$.

As before we restrict attention to those policies that maintain $x(t) \in B^0 \equiv [0,1]$ for all $t > 0$ and to initial states $x \in B^0$.

The first thing we do is to reduce this singular, stochastic control problem into a problem in differential equations. The following result is fairly standard, and follows from an application of Itô's formula. See, for example, [30].

Lemma 1.1. If we can find a twice continuously differentiable function f^*: $B^0 \to \mathbf{R}$ such that $f^*(x) = J(x, L, U)$ for all $x \in B^0$ for some admissible (L, U), and satisfies

$$\min\left(\frac{\sigma^2}{2} f^*_{xx}(x) + b f^*_x(x) - \lambda f^*(x) + h(x), f^*_x(x) + c, -f^*_x(x) + r\right) = 0 \tag{1.3}$$

in B^0, then the (L, U) must be optimal. (Here and elsewhere in this volume f_x denotes the derivative of f and f_{xx} the second derivative.)

Although this theorem has allowed us to translate the problem to one in ODE's it has not yet given us a clue as to its solution. Now suppose we can find a function f in the following class F. Each function $f \in F$ is specified by an interval $B = [b_l, b_u] \subset B^0$, is continuously differentiable in B^0, twice continuously differentiable in the interior of B, and satisfies the following ordinary differential equation (ODE) in B.

$$\frac{1}{2}\sigma^2 f_{xx}(x) + b f_x(x) - \lambda f(x) + h(x) = 0, \tag{1.4}$$

$$f_x(b_l) = -c, \quad \text{and} \tag{1.5}$$

$$f_x(b_u) = r. \tag{1.6}$$

Furthermore, the function is defined in $B^0 - B$ by the linear extension

$$f(x) = f(b_l) + c(b_l - x) \text{ for } b^0_l \le x < b_l \quad \text{and} \tag{1.7}$$

$$f(x) = f(b_u) + r(x - b_u) \text{ for } b^0_u \ge x > b_u. \tag{1.8}$$

Note that every $f \in F$ is almost a candidate for f^*. The differentiator among these functions $f \in F$, that is, among the intervals B is the need for f^* to be twice continuously differentiable over B^0 and not

just B. This means that the optimal choice of interval B^* will have to be made so as to ensure *smooth pasting* [4] of the solution of the ODE inside B^* and the linear extension outside B^*. Thus we are faced with the problem of finding the solution of an ODE and the domain over which it must be solved simultaneously, resulting in the so-called *free-boundary* problem. The second issue that we need to tackle is whether f^* is a value function under some admissible controls.

Before we go into a procedure for finding B^*, we first note the following connection between functions in F and the so-called *regulated Brownian motions*, which provides us with an interpretation of $f \in F$. Consider policies that maintain x in B with the minimum amount of pushing required to do so. That is, for any interval $B \subset B^0$, let L_B and U_B be the (unique) non-negative, nondecreasing, RCLL processes adapted to $\mathcal{F}_t$ such that

$$\begin{gathered} x(t) \in B \quad \text{for all } t > 0 \\ \int_0^t (b_l - x(s))^+ dL_B(s) = \int_0^t (x(s) - b_u)^+ dU_B(s) = 0 \quad \text{for each } t > 0 \\ U_B(0+) = (x(0) - b_u)^+ \quad \text{and} \quad L_B(0+) = (b_l - x(0))^+. \end{gathered} \tag{1.9}$$

The x that results from the use of such L_B and U_B is called a regulated Brownian motion or *two-sided regulator* applied to Brownian motion. The following result, which is also standard, gives the connection between regulated Brownian motions and $f \in F$.

Lemma 1.2. Consider a $B \subseteq B^0$ and the $f \in F$ corresponding to B that satisfies (1.4–1.6). Then

$$f = E_x \left[\int_0^\infty e^{-\lambda t} h(x(t)) dt + \int_0^\infty e^{-\lambda t} c \cdot dL_B(t) + \int_0^\infty e^{-\lambda t} r \cdot dU_B(t) \right],$$

where L_B and U_B are the unique admissible controls that satisfy (1.9).

So we can search for an f^* by searching for a B^*. Once we find a B^* such that the solution to (1.4–1.6) is twice continuously differentiable on B^0, we are done because the optimal policy is specified by the regulator (L_{B^*}, U_{B^*}). An interpretation that follows from the definition of

regulated Brownian motion is that B^* can be though of as the *region of inaction*. As long as x is inside B^* no controls are applied. Controls are only applied on the boundary of B^*. Thus our search reduces to the search for a region of inaction. We now describe an iterative procedure for finding B^*.

We begin the iterative procedure with B^0 as the initial choice for the region of inaction. We solve the set of Equations (1.4)–(1.6) to find the value function corresponding to a regulated Brownian motion whose region of inaction is B^0. Then we iterate as follows to obtain successive regions of inaction $B^1, B^2, \ldots$ and the corresponding value functions $f^1, f^2, \ldots$. The key to getting this procedure to work to find an update rule that allows to efficiently determine B^{k+1} given B^k and will converge to B^*. With that in mind, we impose two desiderata on the update procedure.

D1 **(The Superset Condition).** We want the regions of inaction to be monotone decreasing; we need $B^{k+1} \subseteq B^k$. As is evident from (1.4)–(1.8), we only obtain "real" information about f^k inside B^k. So we have no way of telling how far to back-out if we did not have a monotone sequence of regions and needed to back out. So the superset condition is a requirement for efficient search.

D2 **(Policy Improvement).** We would like that $f^{k+1}(x) \leq f^k(x)$ for all $x \in B^0$. That is, the policy obtained in the next iteration is an improvement on the current policy. This ensures that the f^k will converge.

Upfront, it is not clear that a procedure that meets both D1 and D2 exists. In what follows we construct such a procedure.

The crucial step to constructing the procedure is deciding on the update rule. Given B^k and f^k, define the right and left second derivatives as $f^{k+}_{xx}(x) := \lim_{\delta \downarrow 0}(f^k_x(x+\delta) - f^k_x(x))/\delta$ and $f^{k-}_{xx}(x) := \lim_{\delta \downarrow 0}(f^k_x(x) - f^k_x(x-\delta))/\delta$. At $x = {b_l}^k$, $f^{k+}_{x_i x_i}(x)$ need not equal $f^{k-}_{x_i x_i}(x)$. (Note that $f^{k-}_{x_i x_i}(x) = 0$ because of our construction.) Consider the case when $f^{k+}_{xx}({b_l}^k) < 0$. The situation is as shown in Figure 1.1.

In this case updating the boundary inwards, i.e., setting ${b_l}^{k+1} > {b_l}^k$ helps us achieve D1 of course. But it also helps us achieve D2.

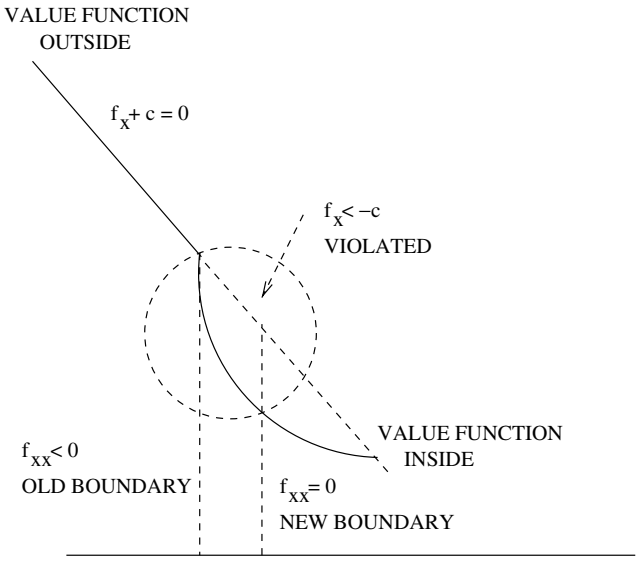

Fig. 1.1 Illustrating the smooth paste update procedure.

To see this, consider a modification ℓ_{B^k} of the control L_{B^k}. Under the control ℓ_{B^k}, if the initial state x is such that $x < {b_l}^k$, it is translated instantaneously, not to ${b_l}^k$ as by L_{B^k}, but to a point x' in the immediate vicinity of ${b_l}^k$ in the interior of B^k. Thereafter, L_{B^k} is mimicked by ℓ_{B^k}. Then

$$J(\ell_{B^k}) - J(L_{B^k}) = \int_{{b_l}^k}^{x'} (f_x^k + c)dx < 0.$$

So ℓ_{B^k} is an improved policy, and by repeating this argument, we suspect that a policy that maintains $x(t)$ in a smaller interval than B^k will be an improvement. (All of this will be formalized shortly.)

Although the heuristic argument above tells us that we should move inwards, it does not tell us by how much. Since we are striving to find a function in $C^2(B^0)$, a natural candidate to update the boundary inwards is the point where $f_{xx}^k(x) = 0$. That is, we move the boundary to a point where, if the resulting region of inaction was indeed the fixed point of the iterations, then the function inside the region (obtained by solving (1.4–1.6)) and its linear extension outside the region would

be smoothly pasted, a requirement for converging to B^* as discussed earlier. Therefore, we choose the local minimizer of f_x^k nearest ${b_l}^k$ as the updated ${b_l}^{k+1}$. That is,

$$\begin{aligned} {b_l}^{k+1} &= \min\{x^* \geq b_l^k | x^* \text{ is a local minimizer of } f_x^k\}, \text{ or equivalently} \\ {b_l}^{k+1} &= \min\left\{x^* \geq b_l^k | \exists\, \epsilon > 0 \text{ s.t. } f_x^k(x^*) = \min_{-\epsilon \leq \delta \leq \epsilon} f_x^k(x^* + \delta)\right\}. \end{aligned} \tag{1.10}$$

Arguing similarly at b_u^k, if $f_{xx}^{k-} < 0$ the natural candidate for the update is

$$\begin{aligned} {b_u}^{k+1} &= \max\{x^* \leq b_u^k | x^* \text{ is a local maximizer of } f_x^k\}, \text{ or equivalently} \\ {b_u}^{k+1} &= \max\left\{x^* \leq b_u^k | \exists\, \epsilon > 0 \text{ s.t. } f_x^k(x^*) = \max_{-\epsilon \leq \delta \leq \epsilon} f_x^k(x^* + \delta)\right\}. \end{aligned} \tag{1.11}$$

For completeness we need to consider the case when $f_{xx}^{k+}({b_l}^k) > 0$. If $f_{xx}^{0+}(0) > 0$ then we suspect that the interval $B^0 = [0,1]$ is not big enough. But our requirement that $x(t) \in [0,1]$ implies that we must simply retain $b_l^1 = 0$. The same holds when $f_{xx}^{0-}(1) > 0$, we must retain $b_u^1 = 1$. Now if these two conditions are ruled out by assumption (by assuming that $c + r$ is sufficiently small, for example) then our update procedure *guarantees* that $f_{xx}^{k+}({b_l}^k) < 0$ implies $f_{xx}^{(k+1)+}({b_l}^k) < 0$. That is, if the procedure works at the first step, it will work at every subsequent step. We illustrate this first via a numerical example taken from [30]. We then quote a result from [30] that the procedure is well-defined and that it converges. The numerical example uses the parameter choices $\lambda = 0.01, \sigma^2 = 2, b = 1, \bar{x} = 0.6, c = 0.02$, $r = 0.01$. Figure 1.2 plots $f^k(x), f_x^k(x), f_{xx}^k(x)$ for $k = 0, 1, 2$.

We start with the initial region of inaction $B^0 = [0,1]$. We then solve the resulting ODE (1.4) in $[0,1]$ with the boundary conditions $f_x^0(0) = -c, f_x^0(1) = r$. Although the resulting expressions are messy, this ODE can be solved analytically. The top set of plots in Figure 1.2 show the resulting f^0 and its first and second derivatives, f_x^0 and f_{xx}^0. As can be seen from the plots, $f_x^0(x) < -c$ for all $x < 0.52$. Thus, (1.3) is violated and therefore we need to move the lower barrier from 0.

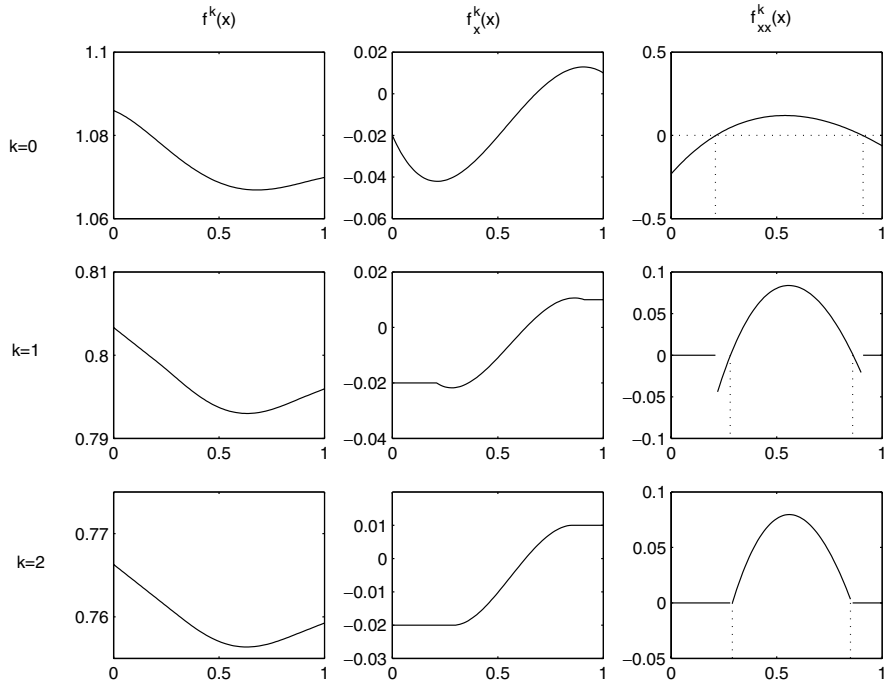

Fig. 1.2 One-dimensional example.

We move the lower boundary to b_l^1 such that $f_x^0(b_l^1)$ is the minimum of f_x^0. Of course, $f_{xx}^0(b_l^1) = 0$. Similarly, we move the right boundary to b_u^1 such that $f_x^0(b_u^1)$ is the maximum of f_x^0. Once again we analytically solve ODE (1.4) for f^1 in $[b_l^1, b_u^1]$ with the boundary conditions $f_x^1(b_l^1) = -c, f_x^0(b_u^1) = r$. The linear extension gives $f^1(x) = f^1(b_l^1) + c(b_l^1 - x)$ for $x \in [b_l^0, b_l^1]$ and $f^1(x) = f^1(b_u^1) + r(x - b_u^1)$ for $x \in [b_u^1, b_u^0]$. The second row of plots in Figure 1.2 show the resulting f^1, f_x^1, and f_{xx}^1. We still have regions where $f_x^1(x) < -c$ and $f_x^1(x) > r$. So we repeat the procedure, arriving at the plots shown in the third row in Figure 1.2. Now we note that $f_x^2(x) \geq -c$ and $f_x^2(x) \leq r$ for all $x \in [0,1]$ and that $f_{xx}^2(b_l^2) = f_{xx}^2(b_u^2) = 0$ upto the resolution of our code, and we terminate the process.

We now provide theoretical justification for the procedure in Proposition 1.3 below, taken from [30].

Proposition 1.3. If $B^k \equiv [b_l^k, b_u^k]$ is such that $f_{xx}^{k+}(b_l^k) < 0$ and $f_{xx}^{k-}(b_u^k) < 0$ for a given k, then

(1) $b_l^{k+1} < b_u^{k+1}$,
(2) $B^{k+1} \subset B^k$, i.e., $b_l^{k+1} > b_l^k$ and $b_u^{k+1} < b_u^k$,
(3) $f^{k+1}(x) < f^k(x)$, for all $x \in B^0$, and
(4) $f_{xx}^{(k+1)+}(b_l^{k+1}) < 0$ and $f_{xx}^{(k+1)-}(b_u^{k+1}) < 0$.

Therefore, if $B^0 \equiv [b_l^0, b_u^0]$ is such that $f_{xx}^{0+}(b_l^0) < 0$ and $f_{xx}^{0-}(b_u^0) < 0$,

(5) $b_l^k \to b_l^*$ and $b_u^k \to b_u^*$, as $k \to \infty$, where $b_l^* > b_l^k$ and $b_u^* < b_u^k$ for any $k \geq 0$,
(6) $f^k \to f^*$ uniformly on B^0 as $k \to \infty$,
(7) $f^* \in C^2(B^0)$ satisfies $\Gamma f^* - \lambda f^* + h = 0$ in $[b_l^*, b_u^*]$, with $f_{xx}^*(b_l^*) = 0$ and $f_{xx}^*(b_u^*) = 0$,
(8) f^* solves (1.3).

Proposition 1.3 establishes several properties of the algorithm. First, it establishes that the update procedure is well-defined, and the region of inactions obtained in each iteration is non-empty. Second, it establishes that update improves the value function D2, and that the superset condition D1 is satisfied. Further, it establishes convergence of the regions of inaction as well as the value function. Finally, it proves that the value function to which the procedure converges is indeed the optimal value function. Moreover, the converged value function can be interpreted as the value function under the policy that instantaneously translates the initial state to $[b_l^*, b_u^*]$ and maintains it thereafter in this interval using the two-sided regulator.

1.2 The General Method

The example in the previous section is quite restrictive. The problem is one dimensional and the stochastic process being controlled is Brownian motion. As a consequence, both identifying the update rule and

verifying its properties is quite easy. Yet, it suffices to provide us with the outline of the general procedure, enumerated below.

(1) Write out the stochastic control problem, as in (1.1), (1.2). In the sections that follow, the control problems we study will be different from the one studied here, but they either will involve singular controls or will be closely related problems such as optimal stopping problems.
(2) Translate the stochastic control problem into a problem in PDEs as in Lemma 1.1. This PDE is commonly called the Hamilton–Jacobi–Bellman (HJB) equation.
(3) Look for solutions to the HJB equation that are specified by free-boundary problems as in (1.4)–(1.8). These correspond to policies that are specified by regions of inaction as in Lemma 1.2. A key issue that arises in higher dimensions is whether classical (smooth) solutions exist for these free-boundary problems. That is, can we find an $f^* \in F$ which is smooth across the boundary of B^*? In some cases, we will quote results that establish existence of smooth solutions. In other cases, we will simply assume existence and proceed. A related issue here is the existence of solutions to the so-called Skorohod problem as in (1.9), which we also simply assume.
(4) We will use an iterative procedure that solves the free-boundary problem by solving a sequence of fixed-boundary problems over domains $B^0, B^1, \ldots$. The key to constructing this procedure is the way to update the boundaries ∂B^k. In every case that we consider, we will insist on D1 and D2 for the reasons mentioned.
(5) At each step in the procedure we will need to solve a PDE over B^k analogous to (1.4) with boundary conditions analogous to (1.5)–(1.6). We need to take into account the direction in which control is exercised in higher dimensions and so the boundary conditions can be nonstandard. The way we set up our method this PDE is a linear elliptic PDE and thus amenable to numerical solution. Although the user is

free to choose the numerical method used to solve this PDE, the ability to handle nonstandard boundary conditions and arbitrary domains B^k suggests the use of Finite Element Methods, which we describe.

(6) In all the cases, we will use the idea of smooth pasting to come up with our update procedure, updating each point on ∂B^k by moving along the direction of control to the point where smooth pasting would hold if the value function did not change, just as in (1.10)–(1.11).

(7) Finally, we need to provide a justification of the method as in Proposition 1.3. In two of the problems we study, namely portfolio optimization with one stock in Section 2 and American option pricing in Section 3, we quote results that provide such guarantees. For the higher dimensional portfolio optimization problems, we offer no guarantees beyond extensive numerical studies.

1.3 Structure and Intended Audience

The intended reader of this survey is one who is conversant with the basic formulations in Mathematical Finance. We provide no modeling justification for the control problems that we study. We assume that the reader knows why these problems are worth solving numerically. We also assume that the reader has at least a passing familiarity with PDE approach to solving these control problems. While we do cite work that carries out the translation from the stochastic control problem to the free-boundary problem in each of our settings, we do not provide any details of this translation. To summarize, this volume is intended for the reader who knows how to set up a stochastic control problem as a free-boundary problem, and wants to find an efficient numerical procedure for its solution. Although the applications we study are in Finance, the actual domain of applicability of our method is much larger. Finally, this survey is intended for those who will actually compute solutions, rather than prove theorems about the control problems or computational schemes. So the level of rigor is not as high as one would find in [20, 27].

One of the shortcomings of this volume is that no general procedure is outlined. The primary reason for this is that it is nearly impossible to justify the method, let alone guarantee its performance, in a sufficiently general set-up that encompasses all the applications we cover. So we choose a case-by-case approach. The procedure is carefully constructed and justified in each of the settings we consider. And in two of these settings performance guarantees are provided. It is the authors' belief that it is easier for the reader to tailor the procedure for an application of interest, and to justify it using the special structure of the problem at hand, by understanding how the moving-boundary method works in a variety of settings.

Finally, we do provide actual MATLAB code for one of the settings we consider, the pricing of American options. This is intended to help the potential user get oriented with implementation of our method.

2

Portfolio Optimization with One Stock and Transaction Costs

In this section, we consider the problem of computing the optimal investment and consumption strategy of an investor who trades a single risky asset and incurs proportional transaction costs. Our objective is to maximize the investors expected discounted utility of consumption over an infinite horizon. This problem is of interest in its own right, of course. But for us, the problem is particularly interesting because it has both singular controls, corresponding to transactions, and classical controls, corresponding to consumption.

The investor manages a portfolio consisting of a risk-free asset, hereafter called the bank, and a risky asset, called a stock. The price process of stock is modeled as a geometric Brownian motion. The investor is given an initial position and in time, can choose to either consume money from the bank, buy stock with money in the bank, or add money to the bank by selling stock. Transacting, that is, buying or selling stock, incurs transaction costs proportional to the values of the transactions themselves. The investor obtains utility by consuming money from the bank. We take the utility function to be either the power utility, or the log utility. The investor's objective is to transact

and consume so as to maximize the expected net present value of utility. The investor is allowed to trade in continuous time and in infinitesimal quantities.

The model described above, without transaction costs was the focus of Merton's seminal paper [39]. The optimal policy proposed by Merton continuously transacts to hold fixed fractions of total wealth in various stocks and consumes another fixed fraction of wealth. Merton's policy requires that an infinite number of transactions be made in any finite time interval. This suggests that in the presence of transaction costs, Merton's policy would no longer be optimal. With transaction costs, the investor would want to make fewer transactions. In particular, he would make transactions only if the fraction of his stock holding is "sufficiently" far away from Merton's optimal fraction to warrant the transaction. Proportional transaction costs were first considered in [36]. It was conjectured that the optimal policy would be characterized by an interval of inaction, such that the optimal policy would not transact when the fraction of wealth in stock lies in this interval. When the fraction wanders outside the interval the optimal policy would be to buy or sell just enough to push the fraction back into the interval.

The problem with proportional transaction costs is now understood to be a singular stochastic control problem [50]. The Merton problem with proportional transaction costs for the one-stock case was solved by Davis and Norman [15]. In an exhaustive theoretical exposition [49] considered a relaxation of the Davis and Norman problem and used viscosity solution techniques to provide existence and uniqueness results and characterized the regularity of the value function. An excellent review of literature in this area is presented in [53]. For us, these theoretical results allow us to reduce the problem to finding a solution to a differential equation with a free-boundary, along the lines of the previous section. Having done so, we will move on to problems involving multiple stocks.

Most of what follows is taken from [15] and [41]. We will use the set-up from Davis and Norman and describe the computational method from [41].

2.1 Problem Formulation

Here, our market consisting of one risk-free and only one risky asset. The risk-free investment, that is the bank, continuously pays an interest rate $r > 0$. The evolution of $S_0(t)$, the value in the bank, can then be expressed as $S_0(t) = S_0(0)e^{rt}$ or,

$$dS_0(t) = r\, S_0(t)dt. \tag{2.1}$$

The risky investment, that is the stock, has a mean rate of return $\alpha > r$. We take a standard Brownian motion $B = \{B(t) : t \geq 0\}$ on its standard filtered probability space $(\Omega, \mathcal{F}, \mathcal{P})$, as our source of uncertainty. Here $\{\mathcal{F}_t : t \geq 0\}$ is a right continuous filtration of σ-algebras that represents the information revealed by the Brownian motion.

Let $S(t)$ denote the value in stock at time t and σ^2 be the variance of the stock returns. Then,

$$dS(t) = S(t)[\alpha\, dt + \sigma dB(t)]. \tag{2.2}$$

The investor is given an initial position of x dollars invested in the bank and y dollars invested in the stock. A consumption and trading policy must be chosen to maximize the objective. Consumption $c(\cdot)$ can occur only from money in the bank and only in non-negative quantities. We assume that $c(\cdot)$ is adapted to $\mathcal{F}_t$, that is, it is nonspeculative and that it be integrable for any finite t, that is,

$$\mathbf{E}\int_0^t c(s)ds < \infty \quad \forall t \geq 0. \tag{2.3}$$

To model the transaction controls we consider two $\mathcal{F}_t$-adapted processes $L(t)$ and $U(t)$, which are non-negative, nondecreasing, and right continuous with left limits. $L(t)$ represents the cumulative amount of money spent from the bank to buy stock before incurring transaction costs. Similarly $U(t)$ represents the cumulative amount of money obtained from selling stock before incurring transaction costs. Buying and selling stock incurs proportional transaction costs. Let $\lambda_l \geq 0$ and $\lambda_u \geq 0$ be the transaction cost for buying and selling stock, respectively. We will assume that $\lambda_l + \lambda_u > 0$ to avoid trivialities. To be more precise, buying a unit worth of stock will cost $(1 + \lambda_l)$ of wealth from the

bank and selling a unit worth of stock will result in $(1 - \lambda_u)$ of wealth added to the bank.

For the sake of readability in the rest of this section, unless necessary, we will suppress the dependence on time t when denoting the processes $B(t), S_0(t), S(t), c(t), L(t), U(t)$. With consumption and transaction, the controlled evolution of S_0 and S can be described by

$$dS_0 = (r\,S_0 - c)dt - (1 + \lambda_l)dL + (1 - \lambda_u)dU, \tag{2.4}$$

$$dS = S\,[\alpha\,dt + \sigma\,dB] + dL - dU. \tag{2.5}$$

The initial position that the investor starts with is (x, y), that is, $S_0(0^-) = x$ and $S(0^-) = y$.

The solvency region is defined as,

$$\Theta = \{(x, y) \in \mathbf{R}^2 : x + \min((1 + \lambda_l)y, (1 - \lambda_u)y) \geq 0\}.$$

The initial portfolio (x, y) and its future evolution are restricted to lie in Θ, which is the set of points from which the investor can conduct transactions to move to a point of non-negative value in both assets. A consumption-transaction policy (c, L, U) is called *admissible* if S_0 and S given by Equations (2.4) and (2.5) lie in Θ for all $t \geq 0$, that is,

$$\mathbf{P}[(S_0, S) \in \Theta \text{ for all } t \geq 0] = 1. \tag{2.6}$$

Therefore, an admissible policy will ensure that bankruptcy does not occur in finite time. We will use $\mathcal{U}$ to denote the set of all admissible policies. $\mathcal{U}$ is clearly nonempty, since we can construct an admissible policy $(\tilde{c}, \tilde{L}, \tilde{U})$ from any policy (c, L, U) by terminating (c, L, U) at any arbitrary time when the state is still in Θ, and moving all wealth to the risk-free asset.

The utility that the investor obtains from consumption is given by the utility function $u(\cdot)$. We will consider two common utility functions, the log utility function and the power utility function. They are given by

$$\text{Log utility:} \quad u(c) = \log(c), \tag{2.7}$$

$$\text{Power utility:} \quad u(c) = \frac{c^\gamma}{\gamma} \quad \gamma \neq 0, \quad \gamma < 1. \tag{2.8}$$

Here γ is the relative risk aversion coefficient that describes the investor's risk preference. These utility functions are very common in modeling investor's risk preference and belong to a class of functions called the *Hyperbolic Absolute Risk Aversion* functions (HARA). Let $\theta > 0$ be the discounting factor. Then the investor's objective is to choose an admissible consumption–transaction policy (c, L, U) so as to maximize

$$J(x, y, c, L, U) = \mathbf{E}_{x,y} \int_0^\infty e^{-\theta t} u(c)\, dt, \tag{2.9}$$

subject to (2.4)–(2.6). The optimal value function V is defined as,

$$V(x, y) = \sup_{(c,L,U)\in\mathcal{U}} J(x, y, c, L, U). \tag{2.10}$$

2.2 The Value Function and the Free-boundary Problem

The stochastic control problem described in the previous section can be transformed into a partial differential equations (PDE) problem (see [49]). The value function V, defined in (2.10), solves

$$\max[\tilde{\mathcal{L}}V, \tilde{\mathcal{S}}V, \tilde{\mathcal{B}}V] = 0 \tag{2.11}$$

where,

$$\tilde{\mathcal{L}}V \equiv \frac{1}{2}\sigma^2 y^2 V_{yy} + \alpha y V_y + rxV_x - \theta V + \max_c (u(c) - cV_x) \tag{2.12}$$

$$\tilde{\mathcal{S}}V \equiv (1 - \lambda_u)V_x - V_y \tag{2.13}$$

$$\tilde{\mathcal{B}}V \equiv -(1 + \lambda_l)V_x + V_y. \tag{2.14}$$

The first-order maximizing conditions for c in (2.12) are,

$$c = \begin{cases} V_x^{-1} & \text{when } u(c) = \log(c), \\ V_x^{\frac{1}{\gamma-1}} & \text{when } u(c) = c^\gamma/\gamma. \end{cases} \tag{2.15}$$

In the above equations and the rest of the section we use V_x to denote the partial differential of V with respect to x, V_y and V_{yy} to denote the single and double partial differential with respect to y.

It is straight forward to show that the optimal value function is concave and satisfies a scaling property known as the homothetic property, that is, for any $\rho > 0$,

$$\begin{aligned} V(\rho x, \rho y) &= \frac{1}{\theta}\log(\rho) + V(x,y) \quad \text{when } u(c) = \log(c) \\ V(\rho x, \rho y) &= \rho^{\gamma} V(x,y) \qquad\qquad \text{when } u(c) = c^{\gamma}/\gamma. \end{aligned}$$

Due to the concavity of the value function and the homothetic property, it can be argued (as in [15, 49]) that Ω is a cone in $\mathbf{R}^2$. This implies that the optimal transaction boundaries can be characterized by two scalar values b_u and b_l such that the region of inaction (no-transaction region) can be represented by

$$\Omega = \left\{ (x,y) \in \Theta : \frac{x}{y} \in (b_u, b_l) \right\}.$$

If $\frac{x}{y} < b_u$ the optimal action is to sell stock till $\frac{x}{y} = b_u$ and similarly if $\frac{x}{y} > b_l$ the optimal action is to buy stock till $\frac{x}{y} = b_l$. In the region of inaction, it is optimal not to conduct any transaction. We will denote by ∂S and ∂B the boundaries $\frac{x}{y} = b_u$ and $\frac{x}{y} = b_l$, respectively. Figure 2.1 illustrates the various regions.

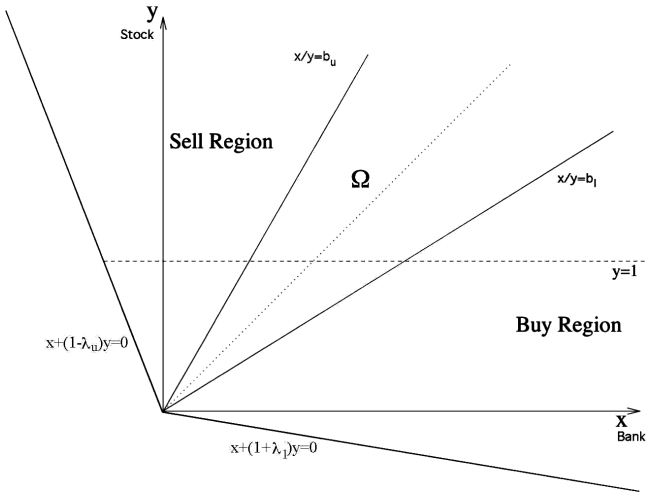

Fig. 2.1 The solvency region.

Furthermore, we can exploit the homothetic property and define a function $W(x)$,

$$W(x) = V(x, 1)$$

to reduce the problem to being one-dimensional. If we can find $W(x)$ $\forall x \in \mathcal{T} \equiv (\lambda_u - 1, \infty)$ we can use the homothetic property to obtain $V(x, y)$, $\forall (x, y) \in \Theta$. For example in the log utility case,

$$V(x, y) = V\left(\frac{x}{y}, 1\right) + \frac{1}{\theta}\log(y) = W\left(\frac{x}{y}\right) + \frac{1}{\theta}\log(y). \tag{2.16}$$

In $\mathcal{T}$, the sell region is $(\lambda_u - 1, b_u)$, the region of inaction $\Omega \equiv (b_u, b_l)$ and the buy region is (b_l, ∞).

In terms of $W(x)$, the HJB equation (2.11) becomes

$$\max[\mathcal{L}W,\, \mathcal{S}W,\, \mathcal{B}W] = 0 \tag{2.17}$$

where,

$$\mathcal{L}W \equiv \beta_3 x^2 W_{xx} + \beta_2 x W_x + \beta_1 W + \beta_4 + \max_c (u(c) - c\, W_x) \tag{2.18}$$

$$\mathcal{S}W \equiv W_x - j_s \tag{2.19}$$

$$\mathcal{B}W \equiv -W_x + j_b. \tag{2.20}$$

For the log utility ($u(c) = \log(c)$) case,

$$\beta_1 = -\theta \tag{2.21}$$

$$\beta_2 = r - \alpha + \sigma^2 \tag{2.22}$$

$$\beta_3 = \frac{1}{2}\sigma^2 \tag{2.23}$$

$$\beta_4 = \frac{1}{\theta}\left(\alpha - \frac{\sigma^2}{2}\right) \tag{2.24}$$

$$j_s = \frac{1}{\theta(1 + x - \lambda_u)} \tag{2.25}$$

$$j_b = \frac{1}{\theta(1 + x + \lambda_l)} \tag{2.26}$$

and for the power utility ($u(c) = \frac{c^\gamma}{\gamma}$) case,

$$\beta_1 = -\frac{1}{2}\sigma^2\gamma(1-\gamma) + \alpha\gamma - \theta \tag{2.27}$$

$$\beta_2 = \sigma^2(1-\gamma) + r - \alpha \tag{2.28}$$

$$\beta_3 = \frac{1}{2}\sigma^2 \tag{2.29}$$

$$\beta_4 = 0 \tag{2.30}$$

$$j_s = \frac{\gamma W}{1 + x - \lambda_u} \tag{2.31}$$

$$j_b = \frac{\gamma W}{1 + x + \lambda_l}. \tag{2.32}$$

As in the above, W_x and W_{xx} represent the first and second derivatives of W.

Of course, the free-boundary problem has not actually gone away despite the reduction to one dimension. We are still left with calculating b_l and b_u, and it is not clear how to do this from the ODE above. To do this, we impose additional structure on the solution. We insist that in $x \in (b_l, b_u)$, the region of inaction, we must have $\mathcal{L}W(x) = 0$, for $x \geq b_u$, $\mathcal{S}W(x) = 0$ and for $x \leq b_l$, $\mathcal{B}W(x) = 0$. This restriction has the interpretation of insisting on selling at or above b_u, buying at or below b_l and not transacting otherwise. If we can find a pair (b_l, b_u) and a function $W(\cdot)$ that satisfy this additional restriction, then it is guaranteed to satisfy (2.17), and so we would have found an optimal solution. The key difficulty that we need to overcome is simultaneously figuring out both the solution of the ODE and the domain over which it must be solved. This is what we will do.

It must not be lost on the reader that the setup in (2.17) has many similarities with (1.3) of the previous section. The restriction here is also very similar to that of (1.4) of the previous section. But there is a crucial difference that must be pointed out. The operator $\mathcal{L}$ has a maximization over c in it. This term arises because we need to also compute the optimal consumption. So expanding the moving-boundary method to handle consumption is the goal.

2.3 The Moving-boundary Method

We will use a two-step procedure to solve (2.17). We begin by choosing an arbitrary region of inaction, $\Omega^0 \equiv (b_u^0, b_l^0)$. For the transaction policy corresponding to Ω^0, we calculate the optimal consumption c_0 and the associated value function W^0 in step 1. In step 2, we use a boundary update procedure that uses Ω^0 and W^0 to obtain a new region of inaction $\Omega^1 \equiv (b_u^1, b_l^1)$ and thus a new transaction policy. We will repeat the procedure to get a sequence of regions of inaction $\Omega^0, \Omega^1, \Omega^2, \ldots$, corresponding consumptions $c_0, c_1, c_2, \ldots$ and corresponding value functions $W^0, W^1, W^2, \ldots$. As in the previous section, the procedure transforms the free-boundary problem (2.17) into a sequence of fixed-boundary problems which are easier to solve numerically.

We assume that the arbitrarily chosen Ω^0 is large enough so that the optimal region of inaction, Ω^* is a subset of Ω^0. However, if Ω^0 is not large enough, Equation (2.46) will not hold, in which case one can restart the iteration with a larger Ω^0.

Step 1: Given Ω^n we seek c_n and W^n such that they solve

$$\hat{\mathcal{L}} W + \max_c (u(c) - cW_x) = 0 \quad \text{in } \Omega^n \tag{2.33}$$

with boundary conditions

$$\mathcal{S} W = 0 \text{ at } b_u \quad \text{and} \tag{2.34}$$

$$\mathcal{B} W = 0 \text{ at } b_l. \tag{2.35}$$

For notational convenience we let $\hat{\mathcal{L}}$ represent the first four terms of $\mathcal{L}$ in Equation (2.18). Again, for readability we will suppress iteration number n in the subscript when the index is obvious. The maximum is achieved in (2.33) by $c = (W_x)^{-1}$ (log utility) or $c = (W_x)^{\frac{1}{\gamma - 1}}$ (power utility). This makes (2.33) a nonlinear elliptic ODE problem. We will use an iterative scheme to solve (2.33)–(2.35) as follows. Start with a guess value for consumption. Given a consumption $c_{n,m}$

$$\hat{\mathcal{L}} W + (u(c_{n,m}) - c_{n,m} W_x) = 0 \text{ in } \Omega^n \tag{2.36}$$

is a linear elliptic equation and can be solved along with (2.34) and (2.35) to obtain $W^{n,m}$, where $W^{n,m}$ would be the value function given

a transaction policy Ω^n and a consumption $c_{n,m}$. For now, we will not worry about how to solve this ODE. We will postpone that to later in this section.

Once we solve the linear problem we update our consumption with the first-order maximization condition in Equation (2.33), that is,

$$c_{n,m+1} = \begin{cases} [W_x^{n,m}]^{-1} & \text{for log utility and} \\ [W_x^{n,m}]^{\frac{1}{\gamma-1}} & \text{for power utility.} \end{cases} \tag{2.37}$$

As is evident, in the representation $c_{n,m}$, n represents the iteration index of the boundary update sequence, while m represents the index of the consumption iteration. Figure 2.2 shows the iterative procedure. A good guess for the initial consumption $c_{n,0}$ would be the converged consumption in iteration $n-1$ and for $n=1$ a good guess is the Merton consumption fraction, that is, the optimal consumption fraction when

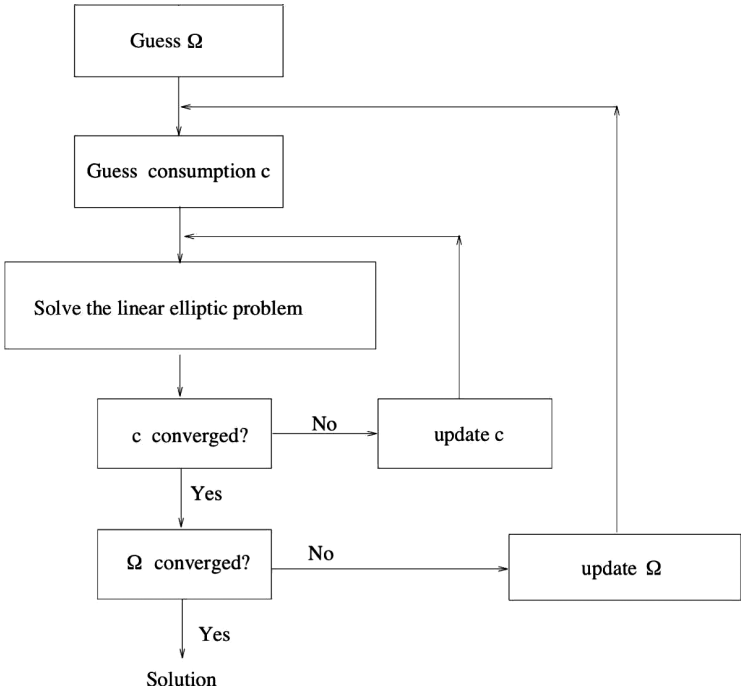

Fig. 2.2 Summary of iteration procedure.

no transaction costs are present,

$$c_{n,0} = \begin{cases} c_{mp} & \text{if } n = 1 \\ c_{n-1} & \text{if } n > 1. \end{cases} \tag{2.38}$$

The Merton consumption c_{mp} is given by,

$$c_{mp} = \begin{cases} \theta(1+x) & \text{for log utility and} \\ \dfrac{1+x}{1-\gamma}\left[\theta - \gamma r - \dfrac{\gamma(\alpha - r)^2}{2\sigma^2(1-\gamma)}\right] & \text{for power utility.} \end{cases} \tag{2.39}$$

We terminate the iterative procedure when

$$\sup_{y \in \Omega^n} |c_{n,m+1}(y) - c_{n,m}(y)| < \epsilon$$

for some tolerance parameter ϵ. In order to implement this procedure we need to solve (2.36) along with (2.34) and (2.35) which constitutes a linear elliptic ODE. Solving such linear elliptic ODEs is straightforward; we will have to say more about this in Section 5.

Step 2: We seek the new region of inaction Ω^{n+1} given Ω^n and W^n calculated in step 1. First, recall from the earlier arguments that,

$$\hat{\mathcal{L}} W^n - (\log(W_x^n) + 1) = 0 \quad \text{in } \Omega^n \tag{2.40}$$

with boundary conditions

$$\mathcal{S}W^n \equiv \; W_x^n - j_s = \; 0 \quad \text{and} \tag{2.41}$$

$$\mathcal{B}W^n \equiv -W_x^n + j_b = 0 \tag{2.42}$$

We also define an operator $\mathcal{Q}$ for notational convenience. The operator $\mathcal{Q}$ maps the set of real valued functions onto the same set. For $z : \mathbf{R} \to \mathbf{R}$,

$$\mathcal{Q}z(x) = \frac{1}{\beta_3 x^2}\big(u(z(x)^{\frac{1}{\gamma-1}}) - \beta_2 x z(x)\big). \tag{2.43}$$

With $\gamma = 0$ for log utility.

If we can create a boundary update sequence that could give us an Ω^{n+1} from $\{\Omega^n, W^n\}$ such that $W^{n+1} > W^n$ and also the assurance that the sequence of Ωs converge, then we have effectively converted the

free-boundary problem into a converging sequence of fixed-boundary problems. Such an update procedure is described by the following equations for b_u^{n+1} and b_l^{n+1},

$$b_u^{n+1} = \inf\{x > b_u^n | (\mathcal{S}W^n)_x = \mathcal{Q}j_s - \mathcal{Q}W_x^n\} \quad \text{and} \tag{2.44}$$

$$b_l^{n+1} = \sup\{x < b_l^n | (\mathcal{B}W^n)_x = \mathcal{Q}j_b - \mathcal{Q}W_x^n\}. \tag{2.45}$$

This is equivalent to moving the boundary $b_u^n(b_l^n)$ toward the interior to the first point where the gradient of $\mathcal{S}W^n$ ($\mathcal{B}W^n$) is equal to $\mathcal{Q}j_s - \mathcal{Q}W_x^n$ ($\mathcal{Q}j_b - \mathcal{Q}W_x^n$).

As in the introductory example discussed in Section 1, the boundary update procedure shown above moves the boundaries in a monotonic fashion. Hence the generated sequence of Ω's are nested, that is, $\Omega^{n+1} \subset \Omega^n$. This makes it obvious that for the boundary update procedure to work we require that our initial guess, Ω^0, contains the optimal no-transactions region Ω^*. For any given Ω^n and W^n, the superset condition which assures us that $\Omega^* \subset \Omega^n$ is,

$$\mathcal{S}W^0\big|_{b_u^0} < \mathcal{S}W^0\big|_{b_u^0+\tau} \quad \text{and} \quad \mathcal{B}W^0\big|_{b_l^0} < \mathcal{B}W^0\big|_{b_l^0-\tau}, \tag{2.46}$$

for some $\delta > 0$ and all $\tau \in (0, \delta)$. The above conditions simply say that it is necessary that the derivative of $\mathcal{S}W^n$ ($\mathcal{B}W^n$) is positive (negative). Therefore, if Ω^0 and W^0 satisfy the above condition, it guarantees that the arbitrarily chosen Ω^0 was large enough. If either of the above conditions fail, then it indicates that the arbitrarily chosen Ω^0 was not large enough. A restart of the procedure with a larger Ω^0 is required. A good way to choose a larger Ω^0 in such cases is to move each boundary half way between the old position and the boundary of the solvency region and check (2.46) again. In the next section we will show that the movement of boundaries dictated by (2.44) and (2.45) is well defined, provided (2.46) is satisfied.

For an arbitrary Ω^n, consider the W^n that is the solution to Equations (2.40)–(2.42). We would expect W^n to violate Equation (2.17). Similar to the introductory example in Section 1, if the superset condition (2.46) holds, then the violation of equation will be in the interior of Ω^n. Moving each boundary to any point in the violation region that is adjacent is bound to yield policy improvement. However, a consideration for where we wish to move is the desire to obtain a Ω^{n+1} that

satisfies the superset condition as well. Such an Ω^{n+1} is provided by the update conditions (2.44) and (2.45).

The stopping criterion for terminating the boundary iterations can be set in two ways — either in terms of the convergence of the region of inaction, that is,

$$\max(b_u^n - b_u^{n-1}, b_l^{n-1} - b_l^n) < \epsilon_b, \tag{2.47}$$

or the convergence of the value function

$$\int_{b_u^{n+1}}^{b_l^{n+1}} \left[W^{n+1}(x) - W^n(x)\right] dx < \epsilon_W. \tag{2.48}$$

Theorem 2.1, stated below and proved in [41], shows that by using the update procedure (2.44) and (2.45), each step of the computational scheme (a) is well defined, (b) results in policy improvement, (c) yields a nested sequence of regions of inaction, and (d) converges to the optimal value function. This is true, provided the initial guess for the region of inaction contains the optimal region of inaction. It considers a function f^n that solves the $\mathcal{L}f^n$ in Ω^n. Provided conditions (2.49) and (2.50) hold, it shows that $\Omega^{n+1} \equiv (b_u^{n+1}, b_l^{n+1})$ is well defined and a subset of Ω^n. If f^{n+1} is the solution to $\mathcal{L}f^{n+1}$ on Ω^{n+1}, Theorem 2.1 also shows that $f^{n+1} > f^n$. Further it shows that conditions (2.49) and (2.50) hold for f^{n+1} as well. This makes it clear that further policy improvement is possible by stepping in, that is, $\Omega^* \subset \Omega^{n+1}$. Equations (2.51) and (2.52) are the simplified forms of Equations (2.44) and (2.45), for the log utility case. Conditions (2.49) and (2.50) are the same as condition (2.46).

Theorem 2.1. Assume that the Merton point is less than 1, that is, $\frac{\alpha - r}{\sigma^2} < 1$. Say, $f^n \in C^2(\Omega^n)$ solves $\mathcal{L}f^n = 0$ in Ω^n with boundary conditions $\mathcal{B}f^n = 0$ at b_l^n and $\mathcal{S}f^n = 0$ at b_u^n. Also say

$$(\mathcal{S}f^n)_x|_{b_u^n +} > 0 \quad \text{and} \tag{2.49}$$

$$(\mathcal{B}f^n)_x|_{b_l^n -} < 0. \tag{2.50}$$

If b_u^{n+1} and b_l^{n+1} are defined by,

$$b_u^{n+1} = \inf\left\{x > b_u^n | (\mathcal{S}f^n)_x = \frac{1}{\beta_3 x^2}\left[\log\frac{f_x^n}{j_s} - \beta_2 x(j_s - f_x^n)\right]\right\} \tag{2.51}$$

and

$$b_l^{n+1} = \sup\left\{x < n_l^n | (\mathcal{B}f^n)_x = \frac{1}{\beta_3 x^2}\left[\log\frac{f_x^n}{j_b} - \beta_2 x(j_b - f_x^n)\right]\right\} \quad (2.52)$$

then

(1) b_u^{n+1}, b_l^{n+1} exist,
(2) $b_u^{n+1} < b_l^{n+1}$ and $\Omega^{n+1} \subset \Omega^n$,
(3) $f^{n+1} > f^n$ in the solvency region $(\lambda_u - 1, \infty)$,
(4) $(\mathcal{S}f^{n+1})_x\big|_{b_u^{n+1}+} > 0$ and $(\mathcal{B}f^{n+1})_x\big|_{b_l^{n+1}-} < 0$,
(5) $b_u^n \to b_u^*$ and $b_l^n \to b_l^*$.

Say f solves $\mathcal{L}f = 0$ in $\Omega^* \equiv (b_u^*, b_l^*)$ with boundary conditions $\mathcal{B}f = 0$ at b_l^* and $\mathcal{S}f = 0$ at b_u^*, then

(6) $f \in C^2(\lambda_u - 1, \infty)$,
(7) $\max(\mathcal{L}f, \mathcal{S}f, \mathcal{B}f) = 0$ in $(\lambda_u - 1, \infty)$,
(8) $f = V$, the optimal value function defined by Equation (2.10).

2.4 Illustrative Results

In this section, we first illustrate the boundary update sequence using an example. We then explore the impact of the transaction cost on the rate of convergence. Both these results are taken from [41]. The goal of these examples is to lay out how the method works. It is not to explore all the possible implications of having a method.

Let us consider a bank paying an interest $r = 7\%$ and a stock with expected rate of return $\alpha = 12\%$ and volatility $\sigma = 0.4$. We take a discounting rate $\theta = 10\%$ and as transaction costs for buying and selling $\lambda_l = \lambda_u = 5\%$. We seek the optimal policy for a risk-averse investor who has the power utility of consumption $(u(c) = c^\gamma/\gamma)$ and a risk aversion coefficient $\gamma = -1$. We begin with a guess transaction policy that buys/sells to maintain the fraction of wealth in stock between 3% and 50%. That is, we do not transact when the percentage of wealth in stock is in the interval $[3, 50]$. The boundary converges in six iterations

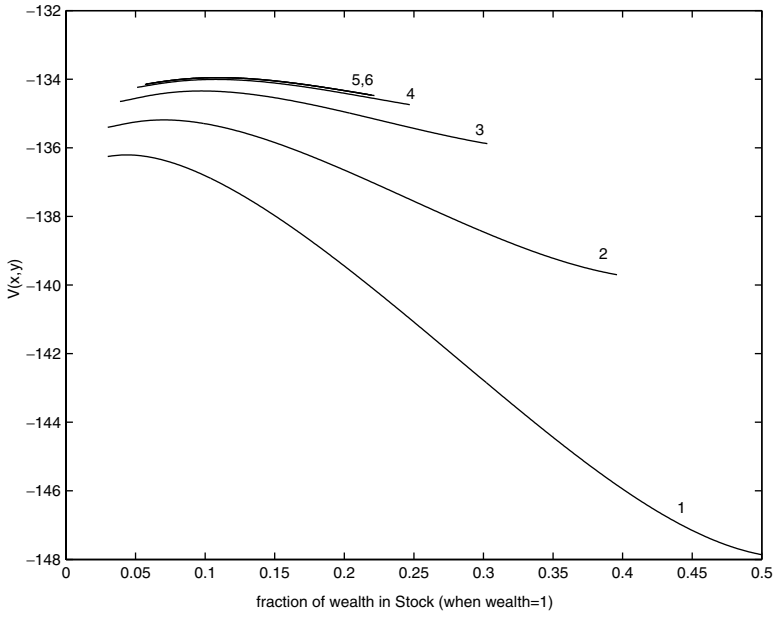

Fig. 2.3 Value functions for each boundary iteration. Illustrating policy improvement.

to within a tolerance of $\epsilon_b = 10^{-2}$. The converged values of b_u and b_l are 5.77% and 21.52%.

Figure 2.3 shows the value function obtained in each iteration. The support of the function is the region of inaction obtained in that iteration. The figure provides a visual illustration of the two properties of Theorem 2.1. First, one can observe that the supports are indeed nested, showing that the regions of inaction obtained are nested as predicted. Second, one can see that the value functions monotonically increase with the iteration. So the method does produce policy improvement. A final thing to note is that convergence is obtained in six iterations. This is commonly observed with the moving boundary method — the number of iterations needed is typically very small.

For various transaction costs ($\lambda = \mu = 0.5\%, 1\%, 2\%, 5\%$, and 10%), Figure 2.4 plots the size of the subsequent Ω^n's (as a percentage of Ω^0, that is $100\frac{\Omega^n}{\Omega^0}$) Vs iteration number. Smaller transaction cost levels correspond to smaller regions of inaction. Hence, it is natural that

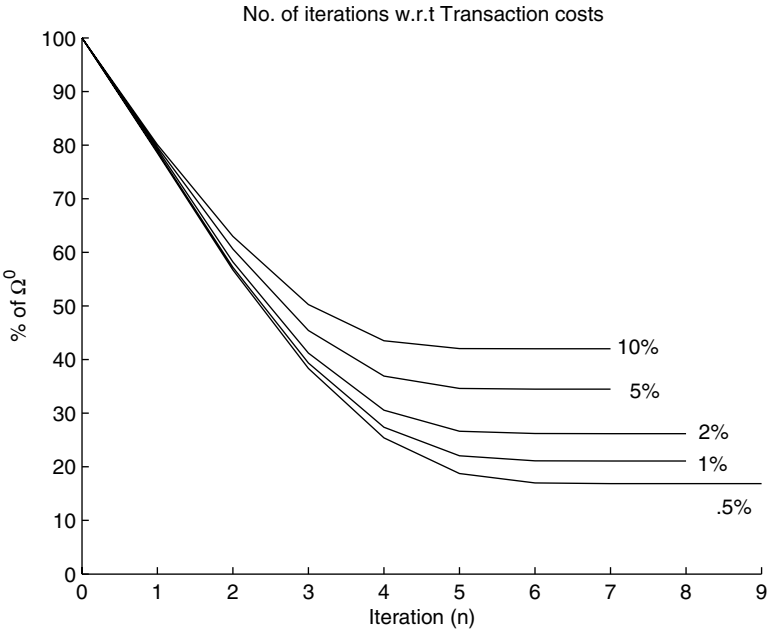

Fig. 2.4 Boundary convergence w.r.t. Transaction levels.

more iterations are required for the convergence of the boundaries when transaction costs become small. But it is interesting to note that the number of iterations required does not significantly increase even for significant changes in transaction levels.

3

American Option Pricing

In the previous two sections we have looked at free-boundary problems that were one dimensional. As a result the boundaries we were computing were simply numbers. In this section, we will study a control problem where the boundary is a function. In that sense we are escalating the difficulty. And, as usual, the problem of computing the price of an American option is interesting in its own right. It is assumed in this section that the reader is familiar with the basics of option pricing. If not, a good place to pick up the requisite background is [35]. Most of this section is taken from [42].

The problem of computing the price of an American option belongs to a class of problems called *optimal stopping* problems where a decision maker decides when to terminate the system in order to maximize a given objective. In the case of American options, termination corresponds to exercise of the option and the objective is to maximize expected pay-off. Calculating the price of the option implicitly involves calculating the optimal exercise policy that a person purchasing the option will use. Thus, pricing involves solving a singular control problem, with the optimal exercise policy that can be characterized by an optimal exercise boundary. Unlike European options,

closed form solutions for the American option price or the optimal exercise boundary do not exist. Hence, an efficient numerical method is very useful.

In the previous sections, we converted a free-boundary problem, where we had to simultaneously determine the solution of a differential equation and the domain over which it must be solved, into a sequence of fixed-boundary problems that were relatively straightforward to solve. That is what we will do here as well. For any given exercise boundary, solving for the value of the option under that policy involves solving a simple PDE with fixed boundaries. Starting with a (nearly) arbitrary boundary and its associated value function, we provide a boundary update procedure to improve the boundary. We show that this method provides a monotone sequence of boundaries that converge to the optimal exercise boundary. In this particular context, it is possible to provide an interesting interpretation of the update procedure — the difference between two consecutive updates is the value of an option with no final pay off but a positive pay-off for an early exercise. And as before, each of the fixed-boundary problems is still linear and can be solved with any standard PDE solver.

From the standpoint of the proposed method, pricing the American call is no different from pricing the American put. Moreover, the price of an American call can be computed using American put pricing methods [37]. The method as well as the proofs of convergence will hold with trivial modifications. Moreover, when the call option is written on a nondividend paying asset, it is well known (see [27]) that it is never optimal to exercise the option early. Thus the price of the American call on a stock that does not pay any dividend is exactly the price of the European call. For these reasons, we restrict our attention to the pricing of American put options only.

3.1 Problem Formulation

Consider an American put option written on an underlying asset. As in the classical setting [6] we will assume perfect markets with a constant risk-free interest rate r and a underlying asset price $S(t)$ given by

$$dS(t) = \mu S(t)dt + \sigma S(t)dB(t).$$

In the above, the constant μ is the expected return, the constant σ is the volatility of the underlying asset price. If q denotes the strike price, then the pay-off at expiration T is $(q - S(T))^+ \equiv \max(q - S(T), 0)$. Of primary interest to us is the price function $p(t,x)$ that represents the price of the put option when the time to expiration is t and the underlying asset price is x. Note here that t represents time to expiration and not simply the time. The option holders objective is to maximize present value of expected payoff from the option. Assuming an absence of arbitrage possibilities, the following facts are well established. We refer the reader to [27] and [45] and the references therein for the verification of these results.

(1) The optimal exercise policy can be represented by a continuous nonincreasing boundary $b(t)$, $0 \le t < \infty$. It is optimal to exercise the put if the current stock price $x \le b(t)$ and hold if $x > b(t)$.

(2) Define the continuation region $\mathcal{C} = \{(t,x) \in (0,\infty)^2;\ x > b(t)\}$ and the exercise region $\mathcal{S} = \{(t,x) \in (0,\infty)^2;\ x \le b(t)\}$. Then $p(\cdot,\cdot)$, $b(\cdot)$ is the unique solution to the free-boundary problem,

$$\mathcal{L}p = 0 \quad \text{in } \mathcal{C} \tag{3.1}$$

$$p(t,b(t)) = q - b(t) \quad t \in [0,\infty) \tag{3.2}$$

$$p(0,x) = (q-x)^+ \quad x \in [b(0),\infty) \tag{3.3}$$

$$\lim_{x\to\infty} \max_{0\le t\le \bar{t}} |p(t,x)| = 0 \quad \bar{t} \in [0,\infty) \tag{3.4}$$

$$\mathcal{M}p = 0 \quad \text{in } \mathcal{S} \tag{3.5}$$

$$p(t,x) \ge (q-x)^+ \ \ (t,x) \in (0,\infty)^2 \quad \text{and} \tag{3.6}$$

$$\lim_{x\downarrow b(t)} p_x(t,x) = -1 \quad t \in (0,\infty), \tag{3.7}$$

where

$$\mathcal{L}p \equiv \frac{1}{2}\sigma^2 x^2 p_{xx} + (r-\delta)xp_x - rp - p_t \quad \text{and} \tag{3.8}$$

$$\mathcal{M}p \equiv (q-x)^+ - p. \tag{3.9}$$

(3) Further, the partial derivatives p_x, p_{xx}, and p_t exist and are continuous on $\mathcal{C}$. Although the smoothness of p_x across $b(\cdot)$

is not known, it is defined and continuous. The optimal exercise boundary is such that $b(0+) = q$ if $r \geq \delta$ and $b(0+) = qr/(\delta)$ if $r < \delta$. The mapping $t \mapsto p(t,x)$ is nondecreasing, $x \mapsto p(t,x)$ is nonincreasing and convex, $x \mapsto x + p(t,x)$ is nondecreasing and convex.

(4) The perpetual boundary or the value to which $b(t)$ goes to asymptotically is given by

$$\lim_{t\to\infty} b(t) = \frac{\tilde{\gamma}}{\tilde{\gamma}-1} q \tag{3.10}$$

where $\tilde{\gamma} = -\frac{1}{\sigma}[\nu + \sqrt{\nu^2 + 2r}]$ and $\nu = \frac{b}{\sigma} - \frac{1}{2}\sigma$. Hence $b(t) \in (\frac{\tilde{\gamma}}{\tilde{\gamma}-1} q, q]$ for all $t \in [0,\infty)$.

Consider Equations (3.1)–(3.7). In $\mathcal{C}$, we have $\mathcal{L}p = 0$. We also have in $\mathcal{S}$, $p = (q-x)^+$ and thus $\mathcal{L}p < 0$. Hence it is necessary that p satisfies

$$\max(\mathcal{L}p, \mathcal{M}p) = 0 \quad \forall (t,x) \in (0,\infty)^2. \tag{3.11}$$

Since the solution to (3.1)–(3.7) is unique, it can be easily verified that any function which has a continuous partial derivative with respect to x and satisfies (3.11) along with the boundary conditions

$$p(0,x) = (q-x)^+ \;\; x \in [b(0),\infty) \tag{3.12}$$

$$\lim_{x\to\infty} \max_{0\leq t\leq \bar{t}} |p(t,x)| = 0 \;\; \bar{t} \in [0,\infty) \tag{3.13}$$

has to be the price function defined by (3.1)–(3.7). Moreover, the regions in which $\mathcal{L}p = 0$ and $(q-x)^+ - p = 0$ are tight would define the continuation and the stopping regions, respectively. Readers familiar with the variational inequality formulation will note that Equation (3.11) is the variational inequality, rewritten in a form that is convenient for our use.

Thus the problem of computing the exercise policy and the price of an American option has been reduced to the problem of solving (3.11) along with (3.12) and (3.13) for a $b(\cdot)$ and a $p(\cdot,\cdot)$ that has continuous partial derivatives. Say we were given the optimal exercise boundary $b(\cdot)$. Then $\mathcal{L}p = 0$ can be solved in the region above $b(\cdot)$ along with (3.12), (3.13) and $p = (q-x)+$ on $b(\cdot)$. In the region below $b(\cdot)$, the

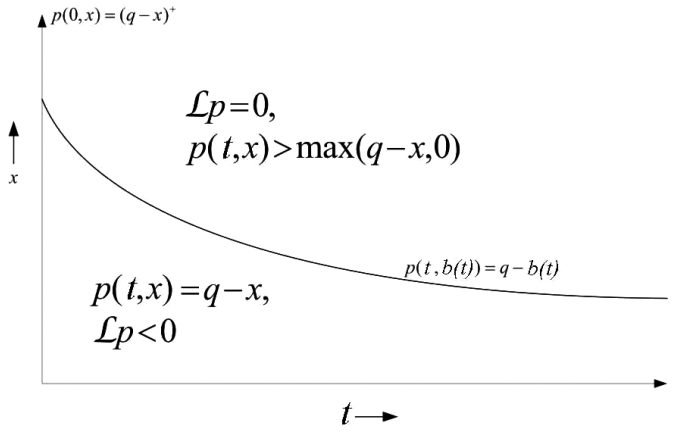

Fig. 3.1 The free-boundary problem.

price p is set to $(q - x)$. Thus obtained p is guaranteed to be the price provided $b(\cdot)$ is the optimal exercise boundary (see Figure 3.1).

As can be noted it would be fairly simple to solve for the price p provided the optimal exercise boundary is known. Any standard PDE solution method or even off the shelf PDE solvers can be used to solve such a PDE problem. However, what makes the problem hard is the need to compute $b(\cdot)$ as a part of the solution itself. This need to jointly solve for both p and b, the solution of a PDE and the domain over which it must be solved, makes the American option pricing problem a free-boundary problem.

3.2 The Moving-Boundaries Approach

Now, consider an arbitrary exercise policy defined by an exercise boundary $b^n(t)$, $0 \leq T \leq \infty$. Since b is continuous, we will consider a b^n that is continuous. Say $p^n(t,x)$ represents the value (price) of the put option if the exercise policy dictated by the boundary b^n is used, that is, exercise the option whenever $x \leq b^n(t)$. Corresponding to our definitions of $\mathcal{C}$ and $\mathcal{S}$, we define $\mathcal{C}^n = \{(t,x) \in (0,\infty)^2;\ x > b^n(t)\}$ and $\mathcal{S}^n = \{(t,x) \in (0,\infty)^2;\ x \leq b^n(t)\}$. An application of Itô's formula yields

that p^n is the solution to the initial-value problem,

$$\mathcal{L}p^n = 0 \quad \text{in } \mathcal{C}^n \tag{3.14}$$

$$p^n(t, b^n(t)) = (q - b^n(t))^+ \quad t \in [0, \infty) \tag{3.15}$$

$$p^n(0, x) = (q - x)^+ \quad x \in [b^n(0), \infty) \tag{3.16}$$

$$\lim_{x \to \infty} \max_{0 \le t \le \bar{t}} |p(t, x)| = 0 \quad \bar{t} \in [0, \infty) \tag{3.17}$$

and in $\mathcal{S}^n$,

$$\mathcal{M}p^n = (q - x)^+ - p^n(t, x) = 0. \tag{3.18}$$

Hence for an arbitrarily chosen boundary b^n, one has at hand a fixed-boundary problem (3.14)–(3.17), (3.18) that can be solved to obtain p^n. By construction, p^n is continuous across b^n. As in the free-boundary problems considered in the previous sections, if b^n is not the optimal boundary, Equation (3.11) would be violated at least in some parts of $(0, \infty)^2$ or p^n_x would not be continuous across the boundary b^n.

Let us consider an example, a put with the following parameters: risk-free rate $r = 8\%$, underlying asset volatility $\sigma = 20\%$ and strike price $q = 100$. Let us pick an arbitrary policy, $b^0(t) = 5$ for all $t \in [0, \infty)$, that is, exercise early only if underlying asset price is or falls below 5. We solve for $p^0(\cdot, \cdot)$ and Figure 3.2 plots a cross section of p^0 taken along the cut $t = 1.5$, that is, $p(1.5, x)$. Now does p^0 satisfy (3.11)? Since b^0 was arbitrarily chosen, we do not expect p^0 to satisfy (3.11). Then where and how is Equation (3.11) violated and can this teach us how b^0 can be improved?

In Figure 3.2, the solid line plots $p^0(1.5, x)$ and the dashed line simply shows $(q - x)$. Figure 3.3 shows $p^0_x(1.5, x)$ which is -1 for $x \le 5$ and then has a discontinuity at $x = 5$. Since $b^0(t) = 5$, $p^0(1.5, x)$ is equal to $q - x$ for $x \le 5$ after which it goes below $(q - x)$, clearly violating Equation (3.11). In this example, the violation occurs between 5 and 93.2. Choosing any point in this region for b^1 would imply policy improvement.

As long as our initial guess b^0 lies below b, in $\mathcal{S}^n$, we would have, from (3.18), $\mathcal{L}p^0 < 0$ and $p^0 = (q - x)^+$. On the other hand, in $\mathcal{C}^0$, Figure 3.3 shows that $p^0_x < -1$ just above b^0. From the boundary condition (3.15), this implies that $\mathcal{M}p^0 = (q - x)^+ - p^0 > 0$ in a region just

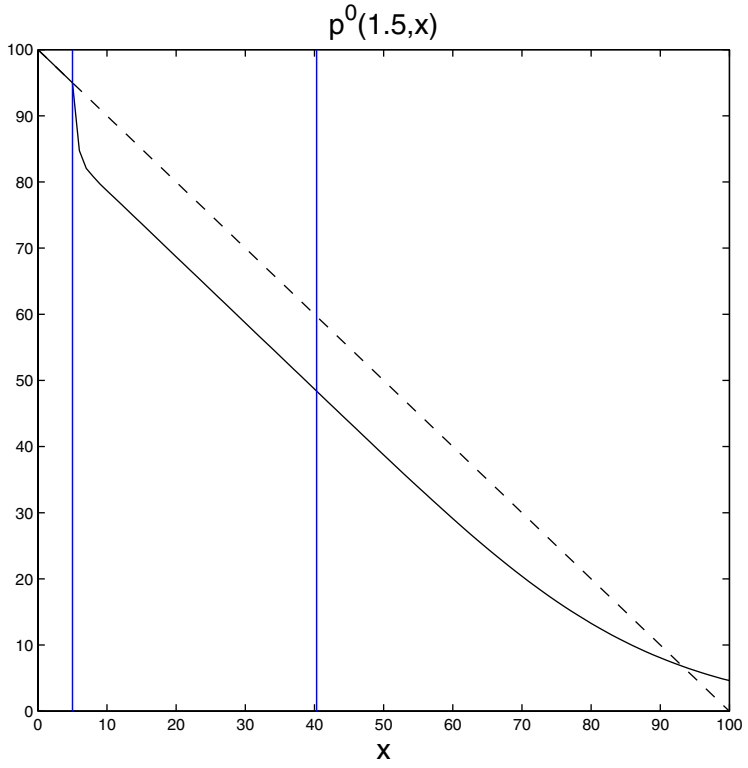

Fig. 3.2 $p^0(1.5, x)$.

above b^0. Hence the violation is in $\mathcal{C}^0$. The violation region is marked as the gray area in Figure 3.4. Like in the free-boundary problems already considered, choosing any boundary b^1 that lies in the violation region will yield policy improvement (Theorem 3.1) but would not necessarily imply that b^1 will still lie below b. However if the new boundary b^1 is chosen as the contour of points that are the first local minima's of $p^0 + x$ along the x direction (Equation (3.19)), then b^1 will satisfy the superset conditions and assure us of policy improvement (Theorem 3.1).

In our example, we choose b^1 as the contour of points on which $p^0 + x$ is minimized. This is equivalent to moving to the point at which $(q - x) - p^0$ is maximized or the point at which p^0_x is -1. The dashed line in Figure 3.3 represents the -1 level, whose intersection with p^0_x

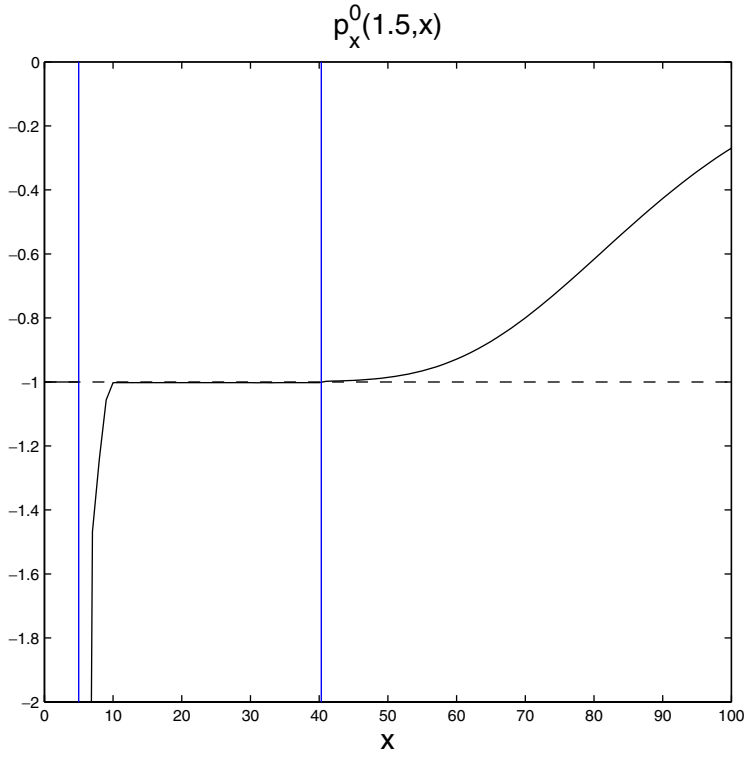

Fig. 3.3 $p_x^0(1.5, x)$.

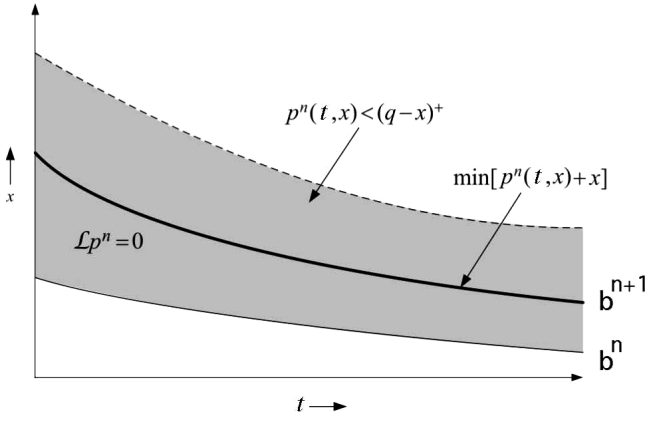

Fig. 3.4 The moving boundary.

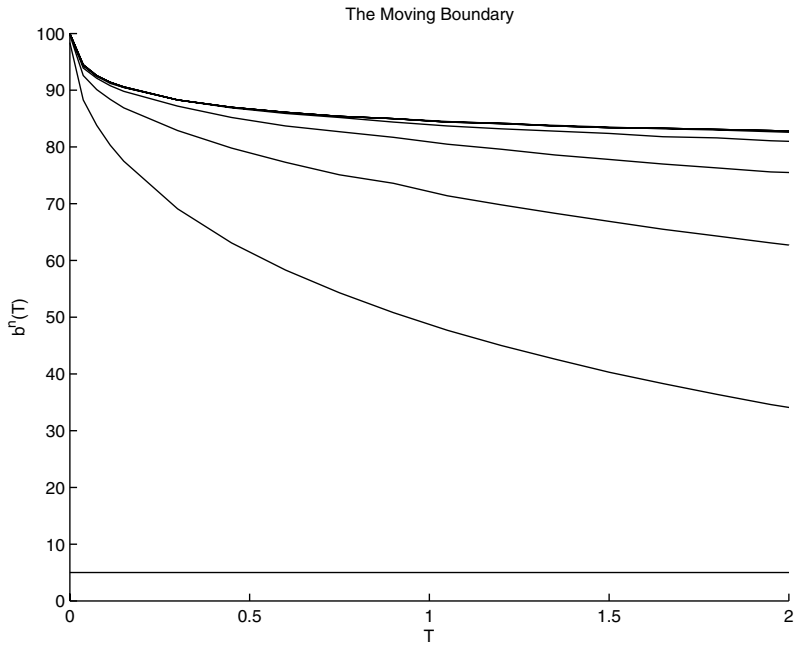

Fig. 3.5 Boundary convergence.

gives the new b^1, which in this case occurs at 40.3 for $t = 1.5$. We pick the new b^1 and iterate.

Figure 3.5 shows the convergence of exercise boundaries. Starting with $b^0(\cdot) = 5$, convergence is attained with $b^6(\cdot)$, that is, in six iterations. A more detailed description of the mechanics of convergence is captured in Figure 3.6. Each row of Figure 3.6 corresponds to one iteration and the first two columns show the value of p^n and p^n_x along the cut $t = 1.5$. The third column shows $b^n(t)$ and $b^{n+1}(t)$ for $t \in [0, 2]$.

The second and third row show the subsequent boundary updates and the corresponding p^n and p^n_x for $n = 1, 2$. The last row shows the plots that correspond to $n = 6$, which is the iteration that results in convergence to within a low tolerance. Here $b^6(1.5) = 83.4$. Note that at convergence, $p^6 \geq (q - x)$ always, $p^6_x = -1$ at b^6 and $p^6_x \geq -1$ always.

The moving-boundary algorithm for American option pricing can be summarized as follows. Begin with a guess b^0 that is guaranteed

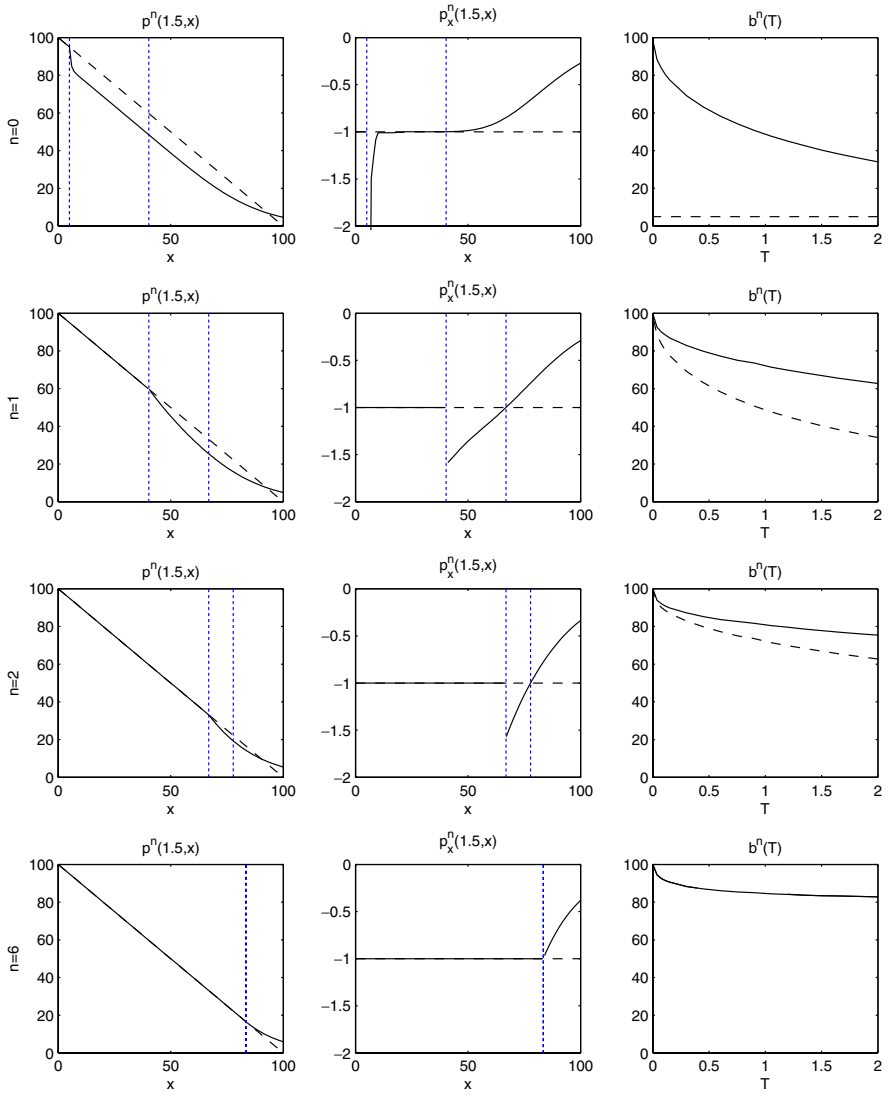

Fig. 3.6 The iterations.

to be below the optimal exercise boundary b, that is $b^0(t) < b(t)$ for all $t \in (0, \infty)$. A convenient choice would be $b^0(t) = b(\infty)$, that is the perpetual boundary given by Equation (3.10) for all $t \in (0, \infty)$. The fixed-boundary problem (3.14)–(3.17) is solved using standard PDE

solvers for p^0. Next, b^1 is obtained from p^0 using the update condition,

$$b^{n+1}(t) = \sup\{x \in (b^n(t), \infty); p_x^n(t, x_0) < -1 \ \ \forall x_0 \in [b^n(t), x)\} \qquad \forall n = 0, 1, \ldots. \tag{3.19}$$

The sequence is then iterated. Theorem 3.1 shows that the moving boundary procedure described above always provides a well defined b^{n+1}, yields a p^{n+1} such that $p^{n+1} > p^n$ and that the sequence b^n converges monotonically. The proof of Theorem 3.1 can be found in [42].

Theorem 3.1. If $p^n \in C^{1,2}$ is the solution to the initial-value problem (3.14)–(3.17) and $p_x^n(t, b^n(t)+) < -1$ for all $t \in (0, \infty)$ then b^{n+1} defined by (3.19) is well defined. Moreover, p^{n+1} (the solution to (3.14)–(3.17) with boundary b^{n+1}) is such that $p^{n+1} > p^n$ and $p_x^{n+1}(t, b^{n+1}(t)+) < -1$ for all $t \in (0, \infty)$.

Theorem 3.1 shows that provided $p_x^n < -1$, we can obtain a b^{n+1} such that the associated p^{n+1} is larger, implying a policy improvement. Moreover, it shows that $p_x^{n+1} < -1$ at $b^{n+1}(t)+$, which allows the repeated use of Theorem 3.1 to obtain a monotone sequence of boundaries each improving the policy. Since the sequence of policies are monotone with an upper bound, convergence is inevitable. It is easy to note that with $p_x = 1$ on the exercise boundary at convergence, the free-boundary problem (3.14)–(3.17) is satisfied.

A heuristic reasoning that provides insight into why the update procedure works can be described as follows. Consider two policies b^n and b^{n+1} and their corresponding price functions p^n and p^{n+1}. The improved policy b^{n+1} is obtained from p^n using Equation (3.19). Now by defining $P = p^{n+1} - p^n$ we note that P solves the same PDE solved by the price functions p^n (3.14) in C^{n+1}. The only difference would be in the boundary conditions satisfied by P. On b^{n+1} instead of (3.15), we would have $P(t, b^{n+1}(t)) > 0$ and at $t = 0$ instead of (3.16) we would have $P(0, x) = 0$. Thus P can be interpreted as the price of a virtual contract which pays 0 on expiry and a positive value if the price process strikes the prescribed boundary b^{n+1}. Since the pay-off of this contract is always non-negative, with possible positive pay-offs, the price P is positive, that is $p^{n+1} > p^n$.

Finally some notes on implementation issues. As in any numerical method one is forced to choose a finite domain and impose boundary conditions on the finite boundaries. Since the problem under consideration is an initial-value problem, the truncation of the t-axis is of no consequence. However, the truncation of the x (vertical) axis brings in an approximation, since the boundary condition (3.17) would be imposed at a finite boundary rather than at infinity. For the purposes of numerical implementation, we choose a $\hat{T}$ that encompasses the times of interest to us along with a large enough $\hat{x}$ and consider the following problem.

$$\begin{aligned} \mathcal{L}p^n &= 0 \quad \text{in } \mathcal{C}^n_{\hat{T},\hat{x}} \\ p^n(t,b^n(t)) &= q - b^n(t) \;\; t \in [0,\hat{T}) \\ p^n(0,x) &= (q-x)^+ \;\; x \in [b^n(0),\hat{x}) \\ p^n(t,\hat{x}) &= 0 \;\; t \in [0,\hat{T}), \end{aligned}$$

where $\mathcal{C}_{\hat{T},\hat{x}} = \{(t,x) \in (0,\hat{T}) \times (0,\hat{x}); x > b^n(t)\}$. Standard methods such as the finite difference method and the Finite-Element Method (FEM) can be employed to solve the fixed-boundary problem. Section 5 provides a brief overview of FEMs.

The stopping criterion for terminating the boundary iterations can be set in two ways — either in terms of the convergence of the optimal exercise region $b(\cdot)$, that is,

$$\max_t (b^n(t) - b^{n-1}(t)) < \epsilon_b$$

or the convergence of the value function

$$\max_{t,x} (p^n(t,x) - p^{n-1}(t,x)) < \epsilon_p.$$

Since the price function becomes less sensitive to the changes in boundary as the boundary sequence approaches the optimal boundary, our implementation tests for convergence to within $\epsilon_b = 10^{-4}$. From the point of numerical computation, both stopping criteria are equivalent and yield the same result for sufficiently small values of tolerance.

3.3 Alternative Numerical Approaches, Runtimes, and Accuracy

Primarily due to the absence of a closed form solution, American option pricing has precipitated a plethora of computational methods. Hence it provides a very rich setting under which the performance of the moving-boundary method can be compared against alternative numerical approaches.

Pricing methodologies for American options can be classified into two broad categories. The first set of methods express the price as the expected value of the pay-off under a risk-neutral measure and compute the expectation. The second set of methods express the price as a solution to the free-boundary HJB equation and computes the solution. While the second set of methods solve for the pricing function, the first set of methods calculate the option price for a given time and underlying asset price.

Unlike the evaluation of the expected pay-off, solving the free-boundary problem has two distinct advantages. First, it provides the optimal exercise policy, which is still usually calculated using the computationally expensive binomial tree method or its variants. Second, it provides the complete pricing function, thereby making the dynamic real-time recalculation of the price unnecessary.

In the earliest work on American options [38], a free-boundary problem for the price function and the optimal exercise boundary (i.e., the free-boundary) is derived. Further the price function is also expressed in terms of the optimal exercise boundary. The analysis was further extended in [40] where the properties of the optimal exercise boundary was studied. Numerical methods to solve the free-boundary problem are developed in [7], [13], and [48]. Closely related to the free-boundary problem is the use of variational inequalities, developed in [5], to characterize the price function. Relations of the free-boundary formulation to the variational inequality formulation is discussed in [21].

Front-fixing methods [33, 46, 52] apply a nonlinear transformation to map the unknown boundary on to a predetermined known boundary and solve the resulting nonlinear problem. Penalty methods [46] on the other hand eliminate the free-boundary by adding a nonlinear penalty

term to the PDE. Both these methods boil down to solving a set of nonlinear equations, the computational speed and accuracy of which largely depends on the initial guess, the problem size and the underlying nonlinear solver used.

There are several methods that exploit the representation of the price as the expected pay-off under the risk-neutral measure. The most common among them are the binomial methods, where the price process of the underlying asset is approximated by a binomial lattice. The binomial method was introduced in [14]. Many generalizations and variants exist. Another approach to computing the expectation is to represent the price as the sum of the European option price and an early exercise premium [12, 25, 29] using an integral equation. A related integral equation is first recursively solved to obtain the optimal exercise boundary and then the price is obtained from a direct numerical integration that depends on the boundary. We refer to this method as the integral method. Richardson extrapolation is used in [23] to solve the integral expression. Pricing methods based on a lower bound and upper bound are provided in [8]. Yet another approach is to use simulation to estimate the price as in [10, 34, 51].

An overview of the tools, pricing methodologies, and the challenges involved in American option pricing can be found in [45], [17], and [9]. Most of the numerical algorithms, results and implementation focus on computing the option price for a given time to expiration and underlying stock price. Detailed comparisons of various numerical methods in this group can be found in [1, 8, 26].

To gauge the performance of the moving-boundary method in terms of both runtimes and accuracy, we make comparisons against popular methods that solve the free-boundary problem. Specifically, we consider, the Brennan and Schwartz method [7], the front fixing method, the penalty method, and the integral method. The following comparisons are taken from [42].

We take as exact the solution of the 10,000-step binomial method. We use two implementations, a finite element implementation (using Femlab libraries for Matlab) and a finite difference implementation to solve the fixed-boundary problems that arise from our method. All our

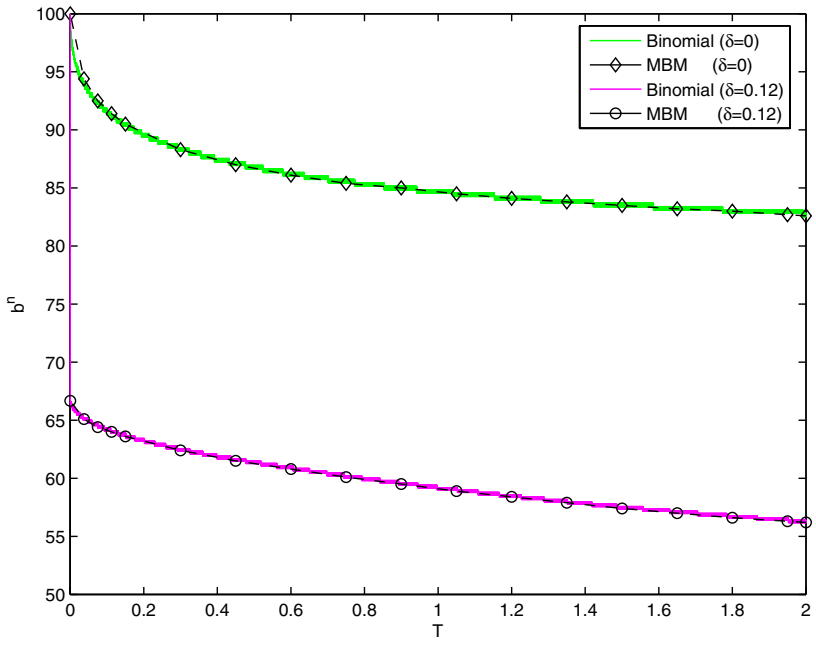

Fig. 3.7 Optimal exercise boundary comparison.

implementations are in Matlab and were run on a 2 Ghz Power Mac G5 machine with 1 GB of RAM.

Figure 3.7 compares the boundaries obtained by the binomial tree method and the proposed method, for $\delta = 0$ and $\delta = 0.12$. Other parameters are kept at $r = 0.08, \sigma = 0.20$, and $q = 100$. Table 3.1 shows $p(3, x)$ obtained by the two methods for $x = 80, 90, 100, 110$, and 120. The values shown in the column titled MBM-FEM of Table 3.1 correspond to the finite element implementation using a 8000 node mesh. Runtime comparisons between the FEM implementation and the binomial tree method are not meaningful since the binomial tree method calculates the price only for a given time to expiration and an underlying stock price, though it explicitly provides the optimal exercise boundary.

In Figure 3.8, we plot runtimes and errors associated with various methods, averaged over 30 different parameter sets. Table 3.1 further shows the corresponding prices for the specific parameter set associated

Table 3.1 Price comparison.

(x,δ)	Binomial	MBM-FEM	MBM-FDM	Bre&Sch	Front-Fixing	Penalty	Integral
(80,0.12)	25.6577	25.6576	25.6407	25.6407	25.6445	25.6803	25.6579
(90,0.12)	20.0832	20.0830	20.0577	20.0577	20.0657	20.1007	20.0833
(100,0.12)	15.4981	15.4973	15.4690	15.4690	15.4733	15.5110	15.4985
(110,0.12)	11.8032	11.8029	11.7741	11.7741	11.7573	11.8114	11.8032
(120,0.12)	8.8856	8.8851	8.8605	8.8605	8.7839	8.8894	8.8855
(80,0)	20.0000	20.0000	20.0000	20.0000	20.0000	20.0000	19.9995
(90,0)	11.6974	11.6969	11.6889	11.6889	11.9029	11.7207	11.6981
(100,0)	6.9320	6.9317	6.9203	6.9203	7.2527	6.9573	6.9325
(110,0)	4.1550	4.1544	4.1427	4.1427	4.4841	4.1760	4.1552
(120,0)	2.5102	2.5098	2.4996	2.4996	2.7760	2.5259	2.5104

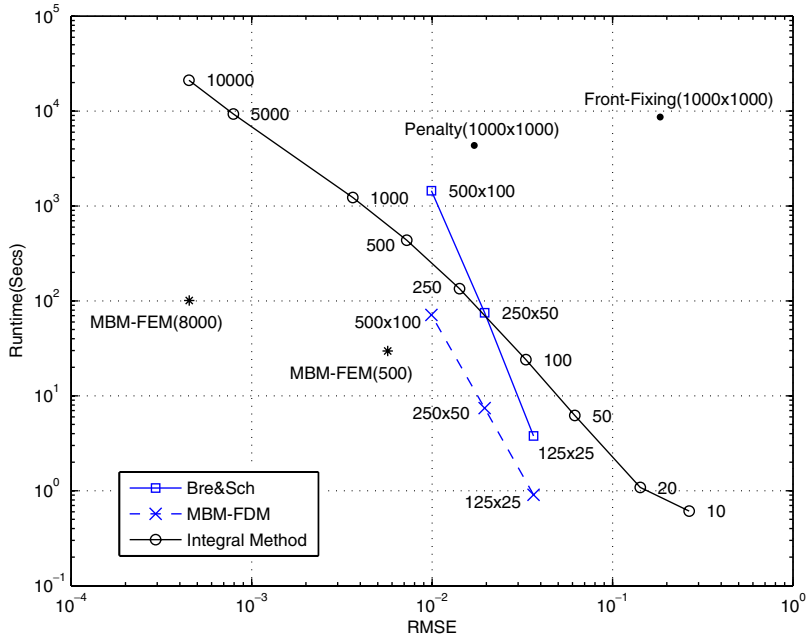

Fig. 3.8 Runtime vs RMSE.

with the table. We discuss below in brief the methods considered and the parameters used. Further details and more comparisons including computational accuracies of greeks can be found in [42].

For the finite element implementation of the moving-boundary method (MBM-FEM). Runtimes and errors associated with two

discrete meshes of 500 and 8000 nodes (generated by the default mesh generation function in Femlab) are shown. All problems were initiated with b^0 set to the perpetual boundary given by Equation (3.10) and converged in less than or equal to six iterations. We use quartic Lagrange elements as the finite element basis.

The finite difference implementation of the moving-boundary method (MBM-FDM) and the implementation of the method presented in Brennan and Schwartz [7] (Bre&Sch), results for three different rectangular mesh sizes of 500×100, 250×50, and 125×25 are plotted. The Brennan and Schwartz method also uses a finite difference scheme. The only difference between the methods is the boundary update procedure and hence the speeds of convergence. Therefore the solution and the RMSEs associated with both these methods are the same (as evident from Table 3.1) while there is considerable speed difference. The values in Table 3.1 correspond to the 500×100 mesh.

The front-fixing method and penalty method use a 1000×1000 rectangular grid and the implicit finite difference scheme described in [46]. As described briefly earlier, this method results in having to solve a system of nonlinear equations. We use the standard nonlinear solver in Matlab's optimization toolbox. One has to keep in mind that the speed and accuracy of such a method largely depends on the ability of the nonlinear solver used. However, irrespective of the solver, large problem sizes become significantly harder to solve relative to linear problems.

Comparison of computational efficiencies of the front-fixing method or the penalty method to the moving-boundary method simply becomes a comparison of solving a set of nonlinear equations as opposed to solving a set of linear equations n times (n is the number of iterations needed, usually ≤ 6).

Integral methods exploit the representation of the American put price as the sum of the European put price (p_0) and an early-exercise premium. The premium is expressed in terms of the optimal exercise boundary. The optimal exercise boundary in turn is expressed as a solution to an integral equation. The integral equation is recursively solved starting with $b(0)$ and progressing in discrete steps. The largest fraction of runtime as well as the significant source of numerical

error arises in solving this equation. Runtimes and errors involved with various discretization frequencies are plotted in Figure 3.8. Values shown in Table 3.1 correspond to the 10,000 step discretization frequency.

Finally, it is important to place the moving-boundary method in the context of the method developed by Kushner and co-workers [31, 32]. Kushner's method is a finite difference discretization scheme for stochastic control problems that allows for the interpretation of the resulting approximation as a Markov Decision Problem (MDP). The resulting MDP can be solved using general techniques like value or policy iteration. This conversion to MDPs helps establish the convergence and the stability of the resulting finite difference approximation. The computational speed and accuracy, however, depend more on the method used to solve the resulting MDP; general schemes that ignore the special structure of singular control problems do not perform as well those that incorporate special structure. The Brennan and Schwartz [7] method, described and compared against as Bre&Sch above, is another finite difference scheme which takes advantage of the underlying problem structure. Since the moving-boundary method outperforms the Brennan and Schwarz method, one can reasonably expect that it will outperform Kushner's method with a general MDP solver for the American Option pricing problem. More importantly, it is not right to think of Kushner's method as a competing method for the moving-boundary method. The moving-boundary technique can be used to solve the MDPs [19] that arise in Kushner's method, as in as in [11], improving the performance over general MDP solvers. A drawback of Kushner's method is that it is limited to finite difference approximations. The moving-boundary method, on the other hand, can be used with either a finite difference (as in MBM-FDM above) or a finite element scheme. The latter provides additional improvement in computational performance, and so one expects that the moving-boundary method combined with finite element discretization will yield performance that is superior to Kushner's method implemented with a general MDP solver. Indeed, this is established via computational experiments on a particular singular control problem in [30].

3.4 Matlab Code for the Finite Difference Implementation

The following MATLAB function computes the price of the American option given $r, \delta, \sigma, K, nx, nt, \hat{X}$, and $\hat{T}$. Here nx and nt represent the number of space and time discretizations, while $\hat{X}$ and $\hat{T}$ represent the values of x and t at which the state space is truncated for numerical purposes. The function returns a matrix P that contains the price of the option at each grid point and a vector b that contains the exercise boundary values for each discrete time point.

The code first pre-computes the finite difference coefficients of the operator $\mathcal{L}$ for all grid points and stores it in matrix A. Then it steps back sequentially in time starting from expiry. For each time point, first a boundary guess bn is made, the corresponding finite difference system of equations are assembled and solved for Pn. If convergence has not been achieved, the boundary is then moved using Equation (3.19) and iterated.

```
function [P,b]=AmericanPrice(r,delta,sigma,K,nx,nt,
               Xhat,That)

%Usage:P=AmericanPrice(r,delta,sigma,K,nx,nt,Xhat,That)
%Example:P=AmericanPrice(0.08,0.12,.2,100,50,10,300,3)

dx=Xhat/nx;
dt=That/nt;
for i=1:nx-1
    A(i,i:i+2)=[((r-delta)*dt*i-sigma^2*dt*i^2)/2...
        1+r*dt+sigma^2*dt*i^2
        (-(r-delta)*dt*i-sigma^2*dt*i^2)/2];
end
P(:,1)=max(K-[0:dx:Xhat],0);

if(delta==0)
    b(1)=K;
else
    b(1)=min(K,K*r/delta);
```

```
end

for j=2:nt+1
    bn=0; run=1;
    while(run)
        An=[A(1+bn:end,1+bn:end)];
        An(end+1,end-1:end)=[-1 1];
        An(end+1,1)=1;
        Cn=[P(bn+2:nx,j-1)' 0 K-bn*dx]';
        Pn=inv(An)*Cn;
        if(Pn(2)$<$K-((bn+1)*dx))
            bn=find(sign(diff(Pn)/dx+1)-1,1,{'last'})+bn;
        else
            b(j)=bn*dx;  run=0;
        end
    end
    P(:,j)=[K-[0:bn-1]*dx Pn'];
end
```

4

Portfolio Optimization with Two Stocks and Transaction Costs

Thus far, we have only discussed problems that have been solved by other methods. Now we come to a problem, where, to the best of our knowledge, only the moving-boundary method has yielded results. Of course, other methods such as that in [32] would work for this problem, but there is no published evidence. And, this problem provides an excellent illustration of how to handle implicit boundary conditions. Most of this section is taken from [43].

The problem we consider here is the two-stock variant of the problem considered in Section 2. As before we will use the homothetic property to reduce the problem to a two-dimensional free-boundary problem, that we will solve by a procedure that is exactly analogous to the one in Section 2. As an illustration of the utility of the moving-boundary method, we will demonstrate the impact of correlation between the two stock returns on the structure of optimal transaction policies.

Although we only consider the two stock case here, the basic formulation is the same when more stocks are concerned. The primary issue with considering more stocks is not difficulty of formulation. Rather, it is the difficulty of obtaining a solution due to the curse

of dimensionality. So for higher dimensions, we will look at a variant of the portfolio optimization problem (without consumption) in Section 6 and describe a moving-boundary method based on approximating the free boundaries by hyperplanes and using simulation to obtain the value function (which is a solution a PDE) at each iteration.

4.1 Problem Formulation

The price of stocks will be represented by a two-dimensional vector S. The evolution of S is modeled analogous to Equation (2.5),

$$dS = \operatorname{diag}(S)\,[\alpha\,dt + \sigma\,dB] + dL - dU, \tag{4.1}$$

where B is a two-dimensional Brownian motion. The investor's initial position is given by $S(0-) = y$, where the ith component of y is represented by y_i. We again define the value function $V(x, y)$ by Equation (2.10), but now take y as a two-dimensional vector. Again as in the one-stock case, one can use dynamic programing arguments in continuous time to show that V solves the free-boundary problem (analogous to Equation (2.11)),

$$\max[\tilde{\mathcal{L}}V, \tilde{\mathcal{B}}_1 V, \tilde{\mathcal{S}}_1 V, \tilde{\mathcal{B}}_2 V, \tilde{\mathcal{S}}_2 V] = 0, \tag{4.2}$$

where

$$\begin{aligned}\tilde{\mathcal{L}}V \equiv\ & \frac{1}{2}\sum_{i=1,2}\sum_{j=1,2} a_{ij} y_i y_j V_{ij} \\ & + \sum_{i=1,2} \alpha_i y_i V_i + (rx - c)V_x - \theta V + u(c) \qquad (4.3)\\ \tilde{\mathcal{B}}_i V =\ & \max_i(-(1+\lambda_{li})V_x + V_i) \qquad (4.4)\\ \tilde{\mathcal{S}}_i V =\ & \max_i((1-\lambda_{ui})V_x - V_i) \qquad (4.5)\\ a =\ & \sigma\sigma' \\ c =\ & \begin{cases} V_x^{\frac{1}{\gamma-1}} & \text{when } u(c) = c^\gamma/\gamma \\ V_x^{-1} & \text{when } u(c) = \log(c). \end{cases} \qquad (4.6)\end{aligned}$$

In the above V_i denotes the partial differential with respect to y_i and V_{ij} denotes $\frac{\partial^2 V}{\partial y_i \partial y_j}$. The value function V is still concave, the region

of inaction is still a cone (now in three dimensions) and homothetic property still holds, that is, for any $\rho > 0$,

$$V(\rho x, \rho y) = \rho^\gamma V(x, y) \quad \text{when } u(c) = c^\gamma/\gamma$$
$$V(\rho x, \rho y) = \frac{1}{\theta}\log(\rho) + V(x, y) \quad \text{when } u(c) = \log(c)$$

As in the one stock case, we can leverage on the homothetic property to reduce the dimensionality of the problem by one, resulting in a free-boundary problem set in two dimensions rather than three. However, to retain symmetry we take an iso-wealth cut rather than the iso-stock cut taken in the one stock case. That is we define,

$$W(y) = V(1 - e \cdot y, y) \tag{4.7}$$

where e is the 2-vector of ones. Figure 4.1 shows the region of inaction and the wealth equals one cut. Figure 4.2 shows the different regions along the wealth equals one cut. The regions outside the region of inaction Ω fall into two categories (Figure 4.2), indicated I and II. In regions

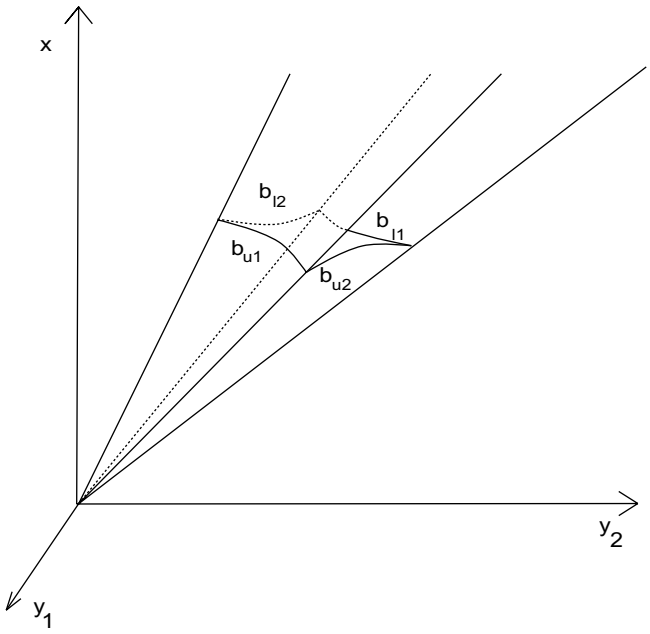

Fig. 4.1 Illustrating homotheticity.

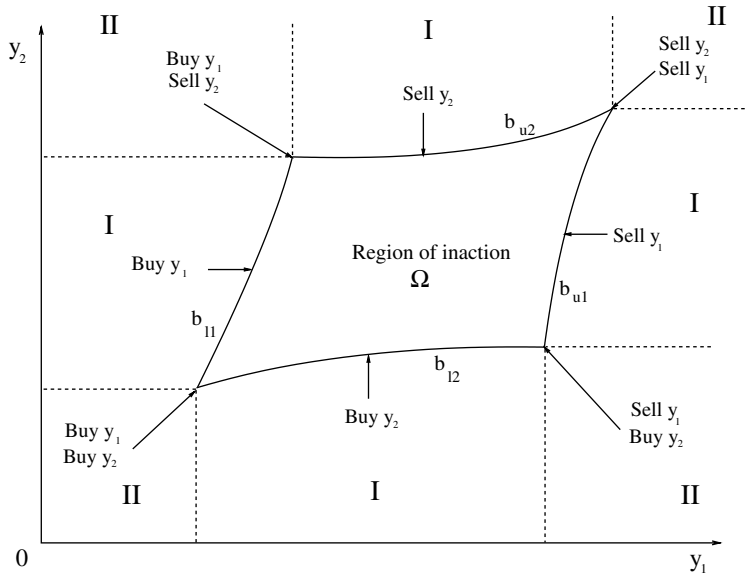

Fig. 4.2 Regions of inaction and transactions.

denoted by I, just enough of one asset is bought or sold to bring the fractions of wealth to Ω. In regions denoted II, it is not possible to transact only one asset to reach Ω. Hence in II, two assets are transacted to reach a corner of Ω. The sequence in which the assets are transacted is irrelevant since transactions are instantaneous. Though the transactions are indicated on the wealth equals one plane, all transactions change the total wealth. So effectively one would be moving to a different wealth plane when transacting.

In this case we have four boundaries. The boundaries that represent buying stock 1 and 2 will be denoted b_{l1} and b_{l2}, respectively. Similarly the boundaries that represent selling stock 1 and 2 will be denoted b_{u1} and b_{u2}, respectively.

The computational procedure for both the log utility and the power utility cases remains essentially the same. Only the specific equations used in the computations differ. Hence we will only consider the case when $u(c) = \log(c)$. In terms of $W(y)$, Equation (4.2) becomes

$$\max\left[\mathcal{L}W,\, \mathcal{B}_1 W,\, \mathcal{S}_1 W,\, \mathcal{B}_2 W,\, \mathcal{S}_2 W\right] = 0 \tag{4.8}$$

with,

$$\mathcal{L}W \equiv \sum_{i=1,2}\sum_{j=1,2}\eta_{ij}W_{ij} + \sum_{i=1,2} b_i W_i - \beta W + \nu + \max_c \left(\log(c) - c\left(\frac{1}{\theta} - \sum_i y_i W_i\right)\right), \tag{4.9}$$

$$\mathcal{B}_j W \equiv W_j\left(y_j + \frac{1}{\lambda_{lj}}\right) - \left(\frac{1}{\theta} - \sum_{i\neq j} W_i y_i\right), \tag{4.10}$$

$$\mathcal{S}_j W \equiv W_j\left(y_j - \frac{1}{\lambda_{uj}}\right) - \left(\frac{1}{\theta} - \sum_{i\neq j} W_i y_i\right), \tag{4.11}$$

where

$$\eta_{ij} = \frac{y_i y_j}{2}\sum_k\sum_l a_{kl}(\delta_{ki} - y_k)(\delta_{lj} - y_l),$$

$$b_i = y_i\left(-\frac{1}{2}\sum_k\sum_l a_{kl}(\delta_{ki}y_l + \delta_{li}y_k - 2y_k y_l) + \sum_k(\delta_{ki} - y_k)(\alpha_k - r)\right)$$

$$\beta = \theta$$

$$\nu = \frac{1}{\theta}\left(r - \frac{1}{2}\sum_k\sum_l a_{kl}y_k y_l + \sum_k y_k(\alpha_k - r)\right).$$

Here δ_{ik} represents the Kronecker delta function, with $\delta_{ik} = 1$ if and only if $i = k$ and $\delta_{ik} = 0$ otherwise.

The reader should note the form of the boundary conditions in (4.10) and (4.11) is not a Dirichlet, Neumann or oblique derivative. Rather, they take the form of *implicit-boundary* conditions. One must adapt the PDE solver to ensure that these conditions can be handled by it. Section 5 provides a brief discussion of finite element methods and how such boundary conditions can be handled.

As with V, using concavity of W we can argue that the solution to (4.8)–(4.11) must reduce to finding W and a region of inaction, Ω^*, such that

$$\mathcal{L}W = 0 \quad \text{in } \Omega^* \tag{4.12}$$

along with $\mathcal{B}_j W = 0$, $\mathcal{S}_j W = 0$ in the buy j regions and the sell j regions, respectively. We would also need that (4.8) holds in the entire solvency region. Using a two step procedure as in Section 2.3, we begin by choosing an arbitrary region of inaction, Ω^0. For the transaction policy corresponding to Ω^0, we calculate the optimal consumption $c^{(0)}$ and the associated value function $W^{(0)}$. In the next step, we use a boundary update procedure that obtains a new region of inaction Ω^1 from Ω^0 and $W^{(0)}$.

We again assume that the arbitrarily chosen Ω^0 is large enough so that the optimal region of inaction, Ω^* is a subset of Ω^0. However, if Ω^0 is not large enough, the superset condition given by Equation (4.21) will fail, in which case one can restart the iteration with a larger Ω^0.

Step 1: For notational convenience, letting $\hat{\mathcal{L}}$ denote the first four terms in Equation (4.9), $c^{(n)}$ and $W^{(n)}$ must solve the nonlinear elliptic PDE,

$$\max_{c^{(n)}} \left[\hat{\mathcal{L}} W^{(n)} + \left(\log(c^{(n)}) - c^{(n)} \left(\frac{1}{\theta} - \sum_i y_i W_i^{(n)} \right) \right) \right] = 0 \tag{4.13}$$

in a given Ω^n, with boundary conditions

$$\mathcal{B}_j W^{(n)} = 0 \text{ at buy } j \text{ boundary}, \tag{4.14}$$

$$\mathcal{S}_j W^{(n)} = 0 \text{ at sell } j \text{ boundary}. \tag{4.15}$$

We will use an iterative scheme to solve (4.13)–(4.15) as follows. Given a consumption $c^{(n,m)}$,

$$\mathcal{L} W^{(n,m)} + \left(\log(c^{(n,m)}) - c^{(n,m)} \left(\frac{1}{\theta} - \sum_i y_i W_i^{(n,m)} \right) \right) = 0 \text{ in } \Omega^n \tag{4.16}$$

is a linear elliptic equation and can be solved along with (4.14) and (4.15) to obtain $W^{(n,m)}$, where $W^{(n,m)}$ is the value function given Ω^n and $c^{(n,m)}$. Once we solve the linear problem we update our consumption with the first-order condition in Equation (4.13), that is,

$$c^{(n,m+1)}(y) = \left[\gamma W^{(n,m)} - \sum_i \left(y_i W_i^{(n,m)} \right) \right]^{-1}. \tag{4.17}$$

In the representation $c^{(n,m)}$, n represents the iteration index of the boundary update sequence, while m represents the index of the consumption iteration. A good guess for the initial consumption $c^{(n,0)}$ would be a value less than the Merton consumption fraction, since with transaction costs one does not expect to consume more than when there are no transaction costs.

$$c^{(n,0)}(y) < \theta \quad \forall y \in \Omega^{(n)}. \tag{4.18}$$

We terminate the iterative procedure when

$$\sup_{y \in \Omega^{(n)}} \left| c^{(n,m+1)}(y) - c^{(n,m)}(y) \right| < \epsilon_c \tag{4.19}$$

for some tolerance parameter ϵ_c.

Step 2: Next we update the region of inaction Ω^n to Ω^{n+1} with b_{bi}^{n+1}'s and b_{si}^{n+1}'s given by

$$\begin{aligned} b_{li}^{n+1} &= \min \big\{ b \geq b_{li}^n | b \text{ is the local maximizer of } \mathcal{B}_i W^{(n)} \\ &\qquad \text{along the } y_i \text{ axis} \big\} \\ b_{ui}^{n+1} &= \max \big\{ b \leq b_{ui}^n | b \text{ is the local maximizer of } \mathcal{S}_i W^{(n)} \\ &\qquad \text{along the } y_i \text{ axis} \big\} \end{aligned} \tag{4.20}$$

Essentially, once W^n is known, each point on the old boundary is moved along the ith axis, toward the interior of Ω^n, to the first point where $\mathcal{B}_i W^{(n)}$ ($\mathcal{S}_i W^{(n)}$) is maximized.

The superset condition that guarantees $\Omega^* \subset \Omega^0$ is,

$$\begin{aligned} \mathcal{B}_i W^{(0)} \Big|_{b_{li}^0} &\leq \mathcal{B}_i W^{(0)} \big|_{b_{li}^0 + \tau e_i} \ \forall i \quad \text{and} \\ \mathcal{S}_i W^{(0)} \Big|_{b_{ui}^0} &\leq \mathcal{S}_i W^{(0)} \big|_{b_{ui}^0 - \tau e_i} \ \forall i, \end{aligned} \tag{4.21}$$

for some $\epsilon > 0$ and all $\tau \in (0, \epsilon)$. The above conditions are higher dimensional versions of (2.46). If either of the above conditions fail, then it indicates that the arbitrarily chosen Ω^0 was not large enough. A restart of the procedure with a larger Ω^0 is required.

Next, we seek insight in three dimensions as to why this update procedure works. Consider the $V^{(n)}$ that can be obtained from $W^{(n)}$ and the step of updating b_{u1}^n to b_{u1}^{n+1}. Figure 4.3 shows the cross-section

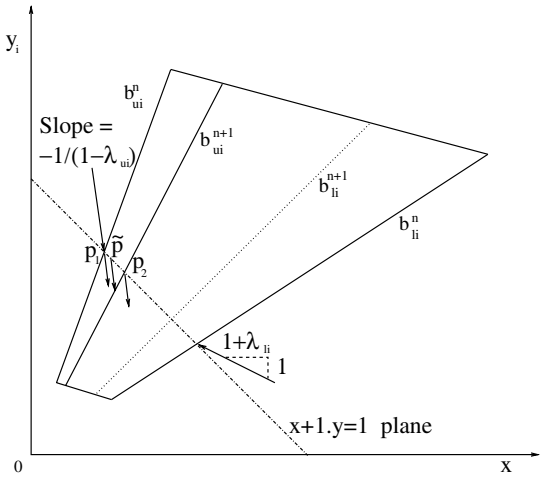

Fig. 4.3 The boundary update procedure.

of the (x, y) space cut at $y_2 =$ constant. Here the x and y axis indicate the wealth in bank and stock 1, respectively. Now say the stochastic processes escapes Ω^n at point $p_1 \in b^n_{s1}$. The transaction policy dictated by Ω^n is that we sell stock i instantaneously. This selling would move the process in the direction $((1 - \lambda_{u1}), -1)$. Also at p_1, we have from the boundary conditions that $\nabla V \cdot ((1 - \lambda_{u1}), -1) = (1 - \lambda_{u1})V_x - V_{y_1} = 0$. Consider a point p very close to p_1 along the iso-wealth line, shown in figure. Say, at p, $(1 - \lambda_{u1})V_x - V_{y_1} > 0$ which indicates that pushing the processes in the sell direction, that is, $((1 - \lambda_{u1}), -1)$ increases the value function V. Since it is optimal to sell stock at p a point arbitrarily close to p_1, it would be profitable to move the boundary to p. Extending the argument, it would be profitable to move the boundary to any point p_3 along the iso-wealth line provided $(1 - \lambda_{u1})V_x - V_{y_1} > 0$ between p_1 and p_3. Figure 4.4 shows a typical plot of $(1 - \lambda_{u1})V_x - V_{y_1}$ along the iso-wealth cut. Moving the left boundary to any point between p_1 and p_3 is bound to yield an improvement in the value function. This suggests that the update procedure is a policy improvement procedure.

We still need to decide how far we need to move in from p_1. As in the one stock case, a consideration for where we wish to move is the desire

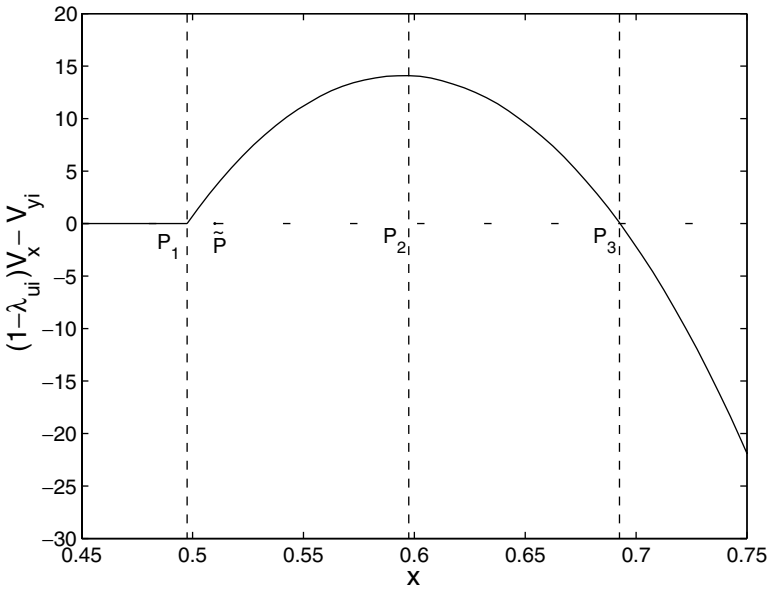

Fig. 4.4 $(1 - \lambda_{u1})V_x - V_{y_1}$ along the transaction cut.

for a procedure that yields a nested sequence of regions of inaction, that is, $\Omega^{n+1} \subset \Omega^n$. If we can choose such an update procedure, this would give us tremendous computational advantage because we no longer need to calculate the value function $W^{(n)}$ in the entire state space and can restrict our attention to calculating it only in the region of inaction Ω^n. Thus, we would have to solve only one PDE rather than $2N + 1$ PDEs at each iteration. To ensure this we pick p_2, that is, the point at which $(1 - \lambda_{u1})V_x - V_{y_1}$ is maximized, rather than p_3.

4.2 Regions of Inaction for the Two-Stock Case

What do the optimal regions of inaction look like? How do they converge? How is the structure of the region of inaction altered when stock returns are correlated? We seek computational answers to these questions in this section.

We begin our exploration by showing a typical boundary update sequence. Later we consider the impact of the covariance structure.

Consider the two-stock problem with the following parameters a bank with interest rate 7%, investors discounting rate 10% and risk aversion coefficient -1, expected returns on both stock 15%, covariance matrix taken as $\begin{pmatrix} 0.4 & 0.1 \\ 0.1 & 0.4 \end{pmatrix}$ with all transaction costs set to 1%. We begin our boundary update procedure with a guess of Ω, as the square, $[0.01, 0.5]^2$, that is, the initial guess for the transaction policy is not to trade when the fractions of wealth in either stock lies in the interval $[0.01, 0.5]$. We use the boundary update procedure and obtain various regions of inaction with each iteration. Figure 4.5 shows the sequence of boundaries generated. Convergence occurs in seven iterations for this case.

Now we turn our attention to the impact of correlation, as an illustration of the utility of our method. In order to make meaningful inferences on the change of shape of the region of inaction, the covariance matrix σ should be changed in a manner so as to keep the Merton point fixed. For the power utility function the Merton point in given by the

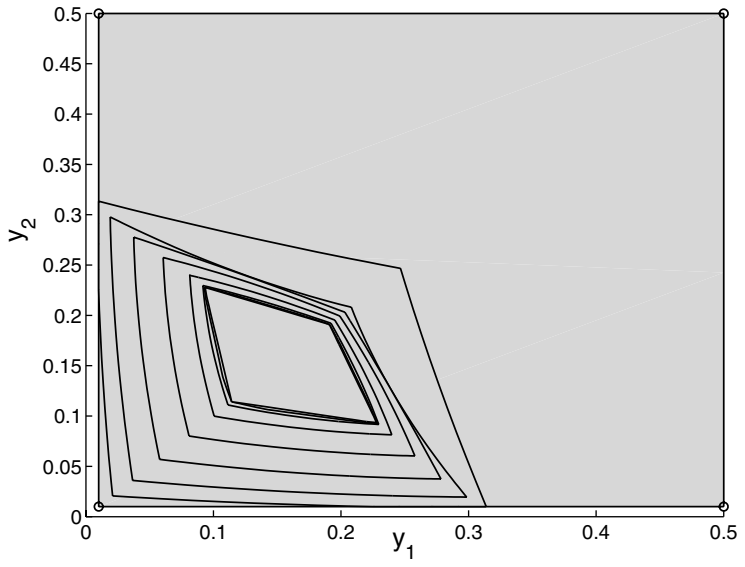

Fig. 4.5 Sequence of boundaries generated.

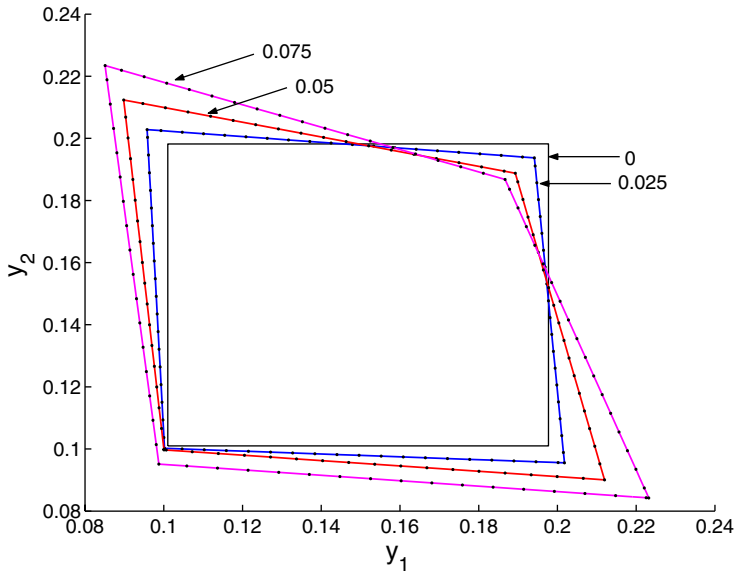

Fig. 4.6 Impact of positive correlation. Stocks with equal expected returns for various values of κ.

vector $\frac{(\sigma\sigma')^{-1}(\alpha-r)}{1-\gamma}$. If $\sigma \equiv \begin{pmatrix} \sigma_v & \sigma_c \\ \sigma_c & \sigma_v \end{pmatrix}$, then the Merton point is

$$\frac{1}{(\sigma_v^2-\sigma_c^2)^2(1-\gamma)} \begin{pmatrix} (\sigma_v^2+\sigma_c^2)(\alpha_1-r) - 2\sigma_v\sigma_c(\alpha_2-r) \\ (\sigma_v^2+\sigma_c^2)(\alpha_2-r) - 2\sigma_v\sigma_c(\alpha_1-r) \end{pmatrix}.$$

If $\alpha_1 = \alpha_2 = \alpha$, say, then the above vector becomes $\frac{1}{(\sigma_v+\sigma_c)^2(1-\gamma)} \begin{pmatrix} (\alpha-r) \\ (\alpha-r) \end{pmatrix}$. The Merton point will not change as long as $\sigma_v + \sigma_c$ remains constant. Therefore Figure 4.6 plots the region of inaction for $\sigma = \begin{pmatrix} 0.4-\kappa & \kappa \\ \kappa & 0.4-\kappa \end{pmatrix}$ for κ varying from 0 to 0.075. When $\kappa = 0.075$ the covariance term is around 23% of the variance term. We retain all the parameters as in the base case but take the expected stock returns for both stock to be 12%.

We observe that the region of inaction shrinks along the $(1,1)$ direction (the main diagonal) and elongates along $(1,-1)$ as the positive correlation, as measured by the parameter κ, increases. With larger values

of κ it is less likely that an increase in the value of stock 1 is accompanied by a decrease in the value of stock 2. Given a region of inaction, it is less likely that sample paths of the value processes will turn away from the main diagonal and hence transactions are more likely to eventually occur along the main diagonal. Given this inevitability, one does not save much on transaction costs by giving the sample paths room to turn away from the boundary along the main diagonal. Therefore, a new region of inaction in which one transacts closer to the Merton point along the main diagonal will provide a better value function because it does not let the value function deteriorate as much before it intervenes. Of course, one can only shrink the region so much along the main diagonal before the transactions costs become prohibitive. The symmetric opposite reasoning explains the elongation along the $(1,-1)$ direction. Another explanation of this behavior is that in the case of positively correlated stock, one does not loose much by having more than the Merton value in one stock and less in the other, since one partially hedges the other. Therefore, one can tolerate more deviation from the Merton point along the $(1,-1)$ direction than the $(1,1)$ direction.

5

Computing the Solution of the Fixed-boundary PDE

In all of the previous sections, we have skirted the issue of solving the PDE that arises in each iteration of our method. This is deliberate. We designed our method in such a way that the PDE in each iteration is a linear elliptic PDE, and there are many ways to solve these PDEs. Any one of them could be used in conjunction with our method. The reader can plug in his or her favorite PDE solver. However, there are three issues to keep in mind. First, we do need good estimates of the gradient of the solution, in order that we can apply our update procedure. Second, we need to solve the problem over arbitrarily shaped domains as our update procedure produces. Third, we will need to deal with implicit boundary conditions such as (4.14) and (4.15).

To accommodate all these three issues we recommend the use of the Finite Element Method (FEM). We describe a version of the FEM that is capable of solving (4.16) with the implicit boundary conditions (4.14) and (4.15). The method is sufficient to solve all the problems studied in this survey. The description of the method is taken from [43]. The reader is advised to consult [24] and [47] for the general treatment of the FEM.

Since we are dealing with just the solution of the PDE in one of the iterations, we will drop the iteration indices (n,m). We will also use $\partial\Omega$ to represent the boundary of the region of inaction Ω. Since c is known in Equation (4.16), for some suitable functions of y: $\tilde{\eta}_{ij}$,$\tilde{\alpha}_i$,$\tilde{\beta}$ and $\tilde{f}$, we can write (4.16) as

$$\sum_i \sum_j \tilde{\eta}_{ij} \psi_{ij} + \sum_i \tilde{\alpha}_i \psi_i + \tilde{\beta}\psi = \tilde{f} \text{ in } \Omega. \tag{5.1}$$

Similarly for some suitable functions r and r_i, the boundary conditions (4.14) and (4.15) can be written as

$$\sum_i r_i \psi_i + r\psi = 0 \text{ on } \partial\Omega. \tag{5.2}$$

Our objective is to find ψ: $\Omega \to \mathbf{R}$ such that Equations (5.1) and (5.2) are satisfied.

First, we write out the so-called *weak* form of (5.1) and (5.2), which is an integral form of Equation (5.1). To solve the weak form of (5.1) we need to find a $\psi \in H^1(\Omega)$ such that for any test function $\vartheta \in H^1(\Omega)$ the following holds

$$\sum_{i,j} \tilde{\eta}_{ij} \left(\int_{\partial\Omega} \psi_i \vartheta - \int_\Omega \psi_i \vartheta_j \right) + \sum_i \tilde{\alpha}_i \int_\Omega \psi_i \vartheta + \tilde{\beta} \int_\Omega \psi \vartheta = \int_\Omega \tilde{f} \vartheta. \tag{5.3}$$

along with the necessary-boundary conditions, going to be described shortly. Here $H^1(\Omega)$ is the Sobolev space of functions that have square integrable generalized first derivatives in Ω, that is, $f \in H^1(\Omega)$ provided,

$$\int_\Omega f_{y_i}^2 dy_i < \infty \quad \forall i.$$

In the usual finite element procedure the Neumann boundary conditions (i.e., ψ_i is known on $\partial\Omega$) are incorporated at this stage by substitution in the first term of (5.3). Dirichlet boundary conditions (ψ is known on $\partial\Omega$) are incorporated by restricting that we find a ψ belonging to $H^1(\Omega)$ and satisfying the Dirichlet boundary condition for all $\vartheta \in H^1(\Omega)$. For our purposes, to account for implicit boundary conditions

we modify the weak form to finding a Λ and $\psi \in H^1(\Omega)$ such that for any two test functions $\vartheta \in H^1(\Omega)$ and $\vartheta_\Lambda \in H^1(\Omega)$ the following holds

$$\begin{aligned}\sum_{i,j} \tilde{\eta}_{ij} \left(\int_{\partial\Omega} \psi_i \vartheta - \int_\Omega \psi_i \vartheta_j \right) \\ + \sum_i \tilde{\alpha}_i \int_\Omega \psi_i \vartheta + \tilde{\beta} \int_\Omega \psi \vartheta + \int_{\partial\Omega} \left[\left(\sum_i r_i \psi_i + r\psi \right) \vartheta_\Lambda - \Lambda \vartheta \right] \\ = \int_\Omega \tilde{f} \vartheta. \end{aligned} \tag{5.4}$$

In (5.4), Λ is an unknown (to be determined) multiplier that is used to include the boundary conditions (5.2) into (5.3).

The following theorem shows that the solution to the strong form always is a solution of the weak form and a solution to the weak form is a solution of the strong form provided it has sufficient smoothness.

Theorem 5.1. (i) If $\psi \in C^2(\Omega)$ solves (5.1) along with (5.2), then ψ solves (5.4) for any two test functions ϑ and ϑ_Λ in $H^1(\Omega)$. (ii) If $\psi \in C^2(\Omega)$ and solves Equation (5.4) for all test functions ϑ and ϑ_Λ in $H^1(\Omega)$, then ψ solves (5.1) along with (5.2).

An issue that we will not address in this survey is whether such a sufficiently smooth ψ exists or not. We will simply assume that it exists and proceed.

The finite element method solves the weak form. As with any numerical scheme the first step is to divide the domain Ω into a collection of simple regions. This is called meshing and can be accomplished easily using one of the many mesh generation routines that are widely available. We use meshes that have a larger mesh density in the neighborhood of the boundaries. Almost all mesh generation programs, like the mesh generation routine in Matlab's PDE toolbox, allow for control of mesh sizes and densities. The FEM along with a well configurable mesh generation routine is well suited to handle domains of arbitrary shapes.

An example of our mesh is shown in Figure 5.1. The vertices of the mesh elements are called nodes. Let M be the total number of nodes

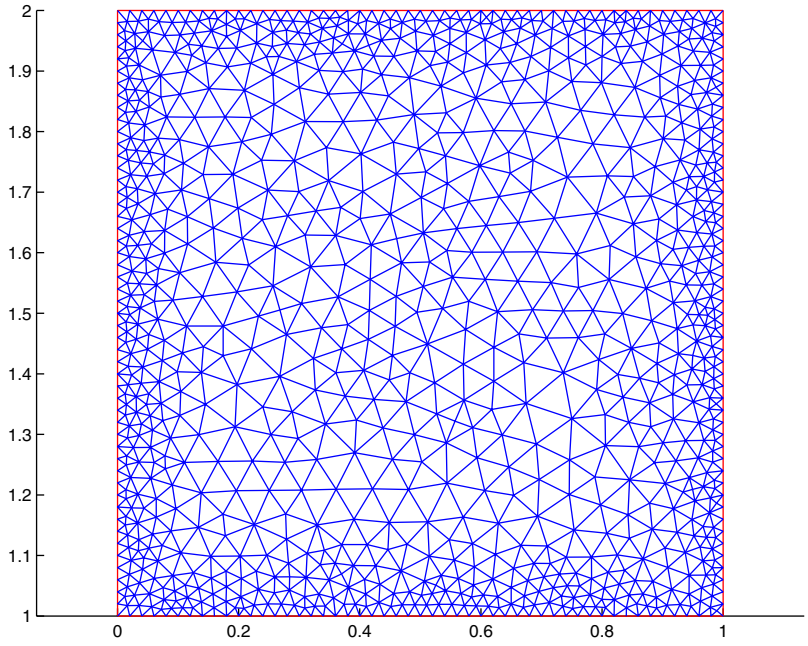

Fig. 5.1 A typical mesh in two dimensions.

and M_p the number of nodes on the boundary. Once we have a mesh, the idea is to introduce an approximation $\hat{\psi}$ to our unknown ψ. Choosing a suitable set of basis functions $\varphi^{(1)}, \varphi^{(2)}, \ldots, \varphi^{(M)}$, we introduce the approximation

$$\hat{\psi} = \sum_l p_l \varphi^{(l)}. \tag{5.5}$$

This transforms the problem of finding the function ψ to finding a $\hat{\psi}$, which is described by a finite number of unknowns p_l. By choosing $\varphi^{(l)}$ to be a function that equals 1 at node l and 0 at all other nodes, p_l will be the value of $\hat{\psi}$ at node l. Similarly we represent Λ by

$$\Lambda = \sum_{l \in \partial\Omega} p_l^{\Lambda} \varphi^{(l)}. \tag{5.6}$$

Note that the notation $\sum_l$ implies that l runs through all nodes, that is $l \in \{i : \text{ Node } i \in \Omega\}$ and $\sum_{l \in \partial\Omega}$ implies that l runs through all boundary nodes only, that is $l \in \{i : \text{ Node } i \in \partial\Omega\}$. Many classes of basis

functions are available and differ by the value they take within the element [24]. Since we seek new boundaries during our boundary iteration step, as functions of the derivative of the value function, we choose Hermite elements as our basis function [24]. Functions that are specified as linear combinations of the Hermite basis functions can be made to have continuous derivatives on the vertices of the elements. This is essential for obtaining a good derivative estimate.

We now discretize the weak form by taking that test functions belong to the space of functions that are linear combinations of our basis functions $\varphi^{(l)}$. This is called the Galerkin method. Thus, representing the test functions by $\vartheta = \sum_k q_k \varphi^{(k)}$ and $\vartheta_\Lambda = \sum_{k\in\partial\Omega} q_k^\Lambda \varphi^{(k)}$, the weak form can be written as

$$q_k \Bigg[\sum_{i,j} \tilde{\eta}_{ij} \left(\int_{\partial\Omega} \psi_i \, \varphi^{(k)} - \int_\Omega \psi_i \, \varphi_j^{(k)} \right) + \sum_i \tilde{\alpha}_i \int_\Omega \psi_i \, \varphi^{(k)} + \tilde{\beta} \int_\Omega \psi \, \varphi^{(k)} - \int_{\partial\Omega} \Lambda \, \varphi^{(k)} - \int_\Omega \tilde{f} \, \varphi^{(k)} \Bigg] + q_k^\Lambda \left[\int_{\partial\Omega} \left(\sum_i r_i \psi_i + r\psi \right) \varphi^{(k)} \right] = 0. \tag{5.7}$$

Since the above holds for all test functions, each of the two terms in the square brackets above should be equal to zero. Approximating the ψ in the first term of (5.7) with $\hat{\psi}$ given by (5.5), and using Equation (5.6) we have,

$$p_l \Bigg[\sum_{i,j} \tilde{\eta}_{ij} \left(\int_{\partial\Omega} \varphi_i^{(l)} \, \varphi^{(k)} - \int_\Omega \varphi_i^{(l)} \, \varphi_j^{(k)} \right) + \sum_i \tilde{\alpha}_i \int_\Omega \varphi_i^{(l)} \, \varphi^{(k)} + \tilde{\beta} \int_\Omega \varphi^{(l)} \, \varphi^{(k)} \Bigg] - p_l^\Lambda \left[\int_{\partial\Omega} \varphi^{(l)} \, \varphi^{(k)} \right] = \int_\Omega \tilde{f} \, \varphi^{(k)}. \tag{5.8}$$

Equation (5.8) has M unknown p_l's and M_p unknown p_l^Λ's. We can write (5.8) as

$$AP = F, \tag{5.9}$$

where

$$A_{kl} = \begin{cases} \sum_i \sum_j \tilde{\eta}_{ij} \left(\int_{\partial\Omega} \varphi_i^{(l)} \varphi^{(k)} - \int_\Omega \varphi_i^{(l)} \varphi_j^{(k)} \right) & \text{for } l \leq M \\ + \sum_i \tilde{\alpha}_i \int_\Omega \varphi_i^{(l)} \varphi^{(k)} + \tilde{\beta} \int_\Omega \varphi^{(l)} \varphi^{(k)} & \\ - \int_{\partial\Omega} \varphi^{(l)} \varphi^{(k)} & \text{for } M < l \leq M_p \end{cases}$$

$$F_k = \int_\Omega \tilde{f} \varphi^{(k)}$$

$$P = \{p_1, p_2, \ldots, p_M, p_1^\Lambda, \ldots, p_{M_p}^\Lambda\}.$$

However as specified A is $M \times (M + M_p)$ and is obviously of not rank $M + M_p$. We need to augment the system with additional constraints so as to uniquely specify P. These constraints come from the second term of (5.7).

$$\int_{\partial\Omega} \left(\sum_i r_i \varphi_i^{(l)} + r\varphi^{(l)} \right) \varphi^{(k)} p_l = 0,$$

which can be written as

$$RP = 0. \tag{5.10}$$

with

$$R_{kl} = \begin{cases} \int_{\partial\Omega} \left(\sum_i r_i \varphi_i^{(l)} + r\varphi^{(l)} \right) \varphi^{(k)} & \text{for } l \leq M \\ 0 & \text{for } M < l \leq M_p. \end{cases}$$

We combine (5.9) and (5.10) to form a combined $(M + M_p) \times (M + M_p)$ system

$$\begin{bmatrix} A \\ R \end{bmatrix} P = \begin{bmatrix} F \\ 0 \end{bmatrix}. \tag{5.11}$$

Thus we solve (5.11) to obtain P. We report $\hat{\psi} = \sum p_l \varphi^{(l)}$ as the solution. Moreover, we report the gradient as $\nabla\hat{\psi} = \sum p_l \nabla\varphi^{(l)}$. Thus we produce all the information that the moving-boundary method needs to do its iterations.

6

Portfolio Optimization with Many Stocks

Although the transformation into a sequence of fixed-boundary problems provides an efficient way to solve such free-boundary problems, it does involve the solution to fixed-boundary problems. When in a higher dimensional setting, it therefore inherits all the difficulties of solving PDEs in higher dimensions. In portfolio optimization problems like the one considered in Section 4 the number of dimensions (independent variables) in the free-boundary problem is equal to the number of risky assets or stock. Since markets usually contain a large number of risky assets, the solution to the portfolio optimization problem set in large dimensions is of natural interest.

In this section, we demonstrate that it is possible to overcome the challenges in higher dimensions, if one is willing to lower aspirations slightly. To this end, we study a variant of the portfolio optimization problem in Section 4. We lower our aspirations and look for the best free-boundary in a class of boundaries that are easily parametrized. Finally, we rely on simulation to obtain an estimate of the value function at certain points in the state space rather than solve the PDE in the entire state space. This section suppresses several details that are

unnecessary for illustrating the main idea. For a detailed discussion we refer readers to [44], from which most of this section is taken.

The problem we consider here is again a continuous time portfolio optimization problem with proportional transaction costs. However, we ignore consumption control and choose the objective of maximizing long-term growth rate of the portfolio. Denoting the portfolio's total wealth over time as a stochastic process $W(t)$, the objective is to maximize

$$\liminf_{t\to\infty} \mathbf{E}\left\{\frac{\log W(t)}{t}\right\}. \tag{6.1}$$

Ignoring consumption in the above objective will allow us to focus on the singular transaction control and at the same time help us demonstrate the handling of average cost objectives rather than discounted objectives that we have always considered.

6.1 Problem Formulation and the Free-boundary Problem

We will consider the same market setup as in Section 4. That is, a market with one risk-free asset and N risky assets modeled by a multi-dimensional geometric Brownian motion. As in Section 4, letting $S_0 \in \mathbf{R}$ and $S \in \mathbf{R}^N$ denote the wealth in the risk-free and risky asset, respectively, we have,

$$dS_0 = r\,S_0\,dt - (e+\lambda_l)\cdot dL + (e-\lambda_u)\cdot dU, \tag{6.2}$$

$$dS = \mathrm{diag}(S)\,[\alpha\,dt + \sigma\,dB] + dL - dU. \tag{6.3}$$

The mean rates of return and the diffusion coefficient are $\alpha \in \mathbf{R}^N$ and $\sigma \in \mathbf{R}^{N\times N}$. The two $\mathcal{F}_t$-adapted RCLL processes $L(t)$ and $U(t)$ model transactions and represent the cumulative amount of money spent and earned in buying and selling the risky assets. The transaction costs are given by $\lambda_l = [\lambda_{l1}, \lambda_{l2}, \ldots, \lambda_{lN}]^T \geq 0$ and $\lambda_u = [\lambda_{u1}, \lambda_{u2}, \ldots, \lambda_{uN}]^T \geq 0$. In order to avoid the trivial case, we again assume that $\sum_i (\lambda_{li} + \lambda_{ui}) > 0$.

The investor's objective is to choose a (L, U) so as to maximize (6.1), where W denotes the total wealth in the portfolio, that is,

$S_0 + \sum_{i=1}^{N} S_i$. Say the expected cumulative cost up to time t when the process X starts from x, can be represented by $(td + V(x))$. Here $V(x)$ is called the differential cost of starting at x or the *differential cost function* and d is the *average expected cost*. Then $\mathbf{E}[\log W(t)]$ can be expanded as

$$\mathbf{E}[\log W(t)] = \log W(0) + rt - (td + V(x)). \tag{6.4}$$

Using which it can be argued that maximizing (6.1) is equivalent to minimizing d. For a detailed discussion on this equivalence refer to [3, 50].

Like in the previous examples considered, classical dynamic programming arguments can then be used as in [50], to show that $V(\cdot), d$ satisfies

$$\min\{\mathcal{L}V(x), (\mathcal{B}_i V(x) | i = 1, \ldots, N), (\mathcal{S}_i V(x) | i = 1, \ldots, N)\} = 0 \tag{6.5}$$

where $\mathcal{B}_i V(x)$ and $\mathcal{S}_i V(x)$ are given by,

$$\begin{aligned} \mathcal{B}_i V(x) &= \lambda_{li} \sum_{j=1}^{N} x_j V_j + V_i + \lambda_{li} \quad \text{and} \\ \mathcal{S}_i V(x) &= \lambda_{ui} \sum_{j=1}^{N} x_j V_j - V_i + \lambda_{ui}. \end{aligned} \tag{6.6}$$

The operator $\mathcal{L}$ is defined by,

$$\begin{aligned} \mathcal{L}V(x) = \nabla V \cdot \big[&\operatorname{diag}(x)(I - ex^T)(\alpha - re - \sigma\sigma^T x) \big] \\ &+ \frac{1}{2} \mathbf{tr}\big\{ D^2 V \operatorname{diag}(x)(I - ex^T)\sigma\sigma^T (I - xe^T) \operatorname{diag}(x) \big\} \\ &+ h(x) - d. \end{aligned} \tag{6.7}$$

The notations $\nabla V, D^2 V$, and $\mathbf{tr}\{\cdot\}$ denote the gradient of V, the Hessian of V and the trace of a matrix, respectively. Again, as earlier, for notational convenience we will write (6.5) as

$$\min\{\mathcal{L}V(x), \quad \mathcal{B}_i V(x), \quad \mathcal{S}_i V(x)\} = 0. \tag{6.8}$$

For uniqueness of the differential cost function we will further restrict $V(0) = 0$. The existence and uniqueness of d as well as the characterization of the differential cost function, V, as the solution (in a viscosity sense) to Equation (6.8) can be found in [2].

Note that in such problems with average cost objectives we seek as solution $V(\cdot)$ and d, with the objective of minimizing d. In discounted pay-off (or cost) objectives considered earlier the unknown was only the value function $V(\cdot)$ that needed to be maximized. The superset condition and the boundary update equation are essentially the same as in Section 4, that is, (4.21) and (4.20). However, the definition of operators $\mathcal{B}$ and $\mathcal{S}$ are in this case given by (6.6).

6.2 A Policy Space Approximation

In this section, we first discuss an appropriate policy space approximation that will facilitate efficient scaling. Then we show how a simulation based variant of the boundary update procedure can be constructed.

A discretization of Ω, represented by a set Π, is a countably finite set such that $x \in \Pi$ implies $x \in \Omega$. The simplest discretization scheme would be to discretize each dimension of Ω^n into P discrete points. This would result in a set Π of size P^N, that is, the number of elements in Π grows exponentially. Moreover, the $2N$ boundaries that completely represent the optimal policy for the N-stock case are each hyper-surfaces. Even the data structure that is required for the representation of a general hyper surface grows exponentially with dimension. Thus, there would be little hope of being able to construct a scheme that scales polynomially with dimension with this discretization.

By restricting the no-transaction regions to polytopes in N-dimensions, we can build a Π that grows polynomially in dimension. No-transaction regions, though not polytopes in general, have been shown to be very close to polytopes [43]. With this approximation we can represent the no-transactions region by

$$AX \geq C, \tag{6.9}$$

where

$$A = \begin{bmatrix} 1 & a^b_{12} & \cdots & a^b_{1N} \\ \vdots & \vdots & \vdots & \vdots \\ a^b_{N1} & a^b_{N2} & \cdots & 1 \\ -1 & -a^s_{12} & \cdots & -a^s_{1N} \\ \vdots & \vdots & \vdots & \vdots \\ -a^s_{N1} & -a^s_{N2} & \cdots & -1 \end{bmatrix} \quad \text{and} \quad C = \begin{bmatrix} b^b_1 \\ \vdots \\ b^b_N \\ -b^s_1 \\ \vdots \\ -b^s_N \end{bmatrix}. \tag{6.10}$$

The elements of A and C describe the buy and sell boundaries. The ith stock's buy and sell boundaries are given by

$$x_i \geq b^b_i - \sum_{j=1\ldots N, j\neq i} a^b_{ij} x_j \quad \text{and} \tag{6.11}$$

$$x_i \leq b^s_i - \sum_{j=1\ldots N, j\neq i} a^s_{ij} x_j, \tag{6.12}$$

respectively. Thus our search for the optimal policy becomes a search for matrix A and vector C.

We first consider the two stock case (Figure 6.1) for the sake of easier description and visualization. To construct Ω^{n+1} from Ω^n, we require the new boundaries $b^{n+1}_{l1}, b^{n+1}_{l2}, b^{n+1}_{u1}$, and b^{n+1}_{u2}. To move b^n_{l1} to b^{n+1}_{l1} using the boundary update Equation (4.20), we seek the local minimizer of $\mathcal{B}\hat{V}_n$. Here $\hat{V}_n$ is a simulation estimate of V_n, details of which will follow in the next section. Since the boundary b_{l1} is approximated by a straight line we would only need two points where $\mathcal{B}\hat{V}_n$ is minimized in order to determine b^{n+1}_{l1}. By discretizing the boundaries b^n_{u2} and b^n_{l2} by P points each, we can obtain the estimate $\hat{V}_n$ on the $2P$ points. However since $\mathcal{B}\hat{V}_n$ depends on the gradient, we would need to estimate $\hat{V}$ on two more lines parallel to b^n_{u2} and b^n_{l2}. Then we can use simple finite differences for gradient estimation. Therefore, to update boundary b^{n+1}_{l1} and b^{n+1}_{u1} for the first stock, we only need to estimate the differential cost function V_n for discretized points on four lines (4P points). We use the estimates on these points to calculate the gradient on two lines (2P points). We will call the lines on which we seek estimates of V_n as well as its gradient as *main lines* and the lines on which we seek estimates of V_n only to facilitate the calculation of the gradients on the main line as

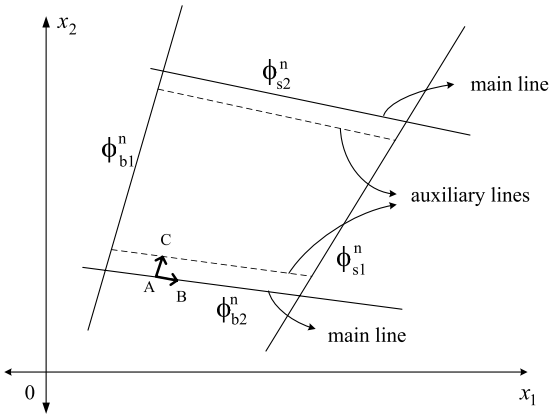

Fig. 6.1 Shape of the no-transaction region in 2-stock case.

auxiliary lines. Note that main lines and accompanying auxiliary lines are parallel. These lines are also shown in Figure 6.1 for the 2 stock case.

Now consider the general N-dimensional case. For a particular stock i, the update of the *buy* i boundary and the *sell* i boundary requires the estimation of $\hat{V}_n$ on N^2 lines. These N^2 lines are in N groups of N lines each. Each group has one main line and $N-1$ auxiliary lines. The gradient is estimated on the N main lines using the estimate of V on all N^2 lines using finite differences as in the two stock case. On each of the N main lines, the boundary update condition (4.20) gives 1 buy point and 1 sell point. Fitting an N-dimensional hyper-plane for the new buy(sell) boundary using the N new buy(sell) points is straight forward. For each of the N stocks we do the same. This implies that the estimation of $\hat{V}$ needs to be done on $N^2 \times N = N^3$ lines. The discretization of these N^3 lines comprises our discretization set Π. If each line is discretized by P points then the size of Π is PN^3, growing polynomially in N.

6.3 Simulation Based Boundary Update

In the previous sections, for any fixed region Ω^n we used PDE methods to solve for V^n. However in order to scale well with dimension, here we

will use simulation to estimate the differential cost function V for only a set of points in Ω^n, but still rely on the boundary update equations obtained by the PDE based arguments to improve policies. When we use simulation we can only obtain estimates of V^n rather than the exact V^n. Since we are dealing only with estimates in each iteration we can no longer be assured that a boundary update based on the V^n estimates will yield a monotone sequence of boundaries and converge to the optimal boundary. However by sequentially increasing the sample paths used in the estimation and by relaxing the superset condition we can construct a modified version of the update procedure that assure convergence to the optimal boundary.

First, let us define some notations. Let M_n be an arbitrarily increasing sequences of positive integers such that $M_n \to \infty$. A discretization of Ω^n, as described in the previous section, will be represented by a set Π^n. Obviously Π^n is countably finite such that $x \in \Pi^n$ implies $x \in \Omega^n$. An estimate of V that uses M_n sample paths will be denoted by $\hat{V}_n^{M_n}$.

In step 1, we start the computation with a guess Ω^0 and $n = 0$. For a given Ω^n, in step 2, using M_n sample paths we obtain an estimate of $V(x)$ (that is $\hat{V}_n^{M_n}$) for each $x \in \Pi^n$. We use standard simulation techniques for the estimations. Since $\hat{V}$ is only an estimate of V, we are not guaranteed that the Ω^n's obtained from update Equation (4.20) always contains Ω^*. We need to check at each step if Ω^n and $\hat{V}^n$ satisfy (4.21), which we do in step 3.

If conditions (4.21) hold, we update the boundaries using (4.20). At this stage, after an inward movement of the boundaries, we define two sets of variables Δb_{li}^n and Δb_{ui}^n by

$$\Delta b_{li}^n = a(b_{li}^n - b_{li}^{n-1}) \quad \text{and} \tag{6.13}$$

$$\Delta b_{ui}^n = a(b_{ui}^n - b_{ui}^{n-1}) \tag{6.14}$$

for some $a \in (0,1)$. Both Δb_{li}^n and Δb_{ui}^n are recalculated only when a boundary update is made using (4.20). Hence they can be interpreted as a fraction a of the last inward movement of the boundaries. The convergence of Ω^* is checked after each inward movement and the iteration is continued if not converged. Convergence can be checked by either testing the convergence of d or Ω^n.

If any of conditions (4.21) fail, then it indicates that (under the accuracy permitted by M_n sample paths) our Ω^n has overshot inwards due to estimation error and a backing out is necessary. We back out by redefining Ω^n. To this extent we first define Ω^{new} by

$$b_{li}^{\text{new}} = b_{li}^n - \Delta b_{li}^n \tag{6.15}$$

$$b_{ui}^{\text{new}} = b_{ui}^n + \Delta b_{ui}^n. \tag{6.16}$$

Upon consecutive backing out it is possible (though rarely) that Ω^{new} (from (6.15) to (6.16)) is no longer a subset of the initial guess Ω^0, in which case we set $\Omega^n = \Omega^0$ otherwise we set $\Omega^n = \Omega^{\text{new}}$. Note that, by our assumption, Ω^* is a subset of Ω^0. Figure 6.2 shows a chart that summarizes the computational scheme.

The idea behind using an increasing number of sample paths to estimate V is to improve on the computational efficiency. Since during the early stages of the iteration Ω^n tends to be relatively further away from Ω^*, the chances of over-shooting due to a cruder estimate of V tends to be lower. Moreover, as $M_n \to \infty$ the scheme itself converges to the boundary update procedure with estimate $\hat{V}$ converging to V. Hence we can get arbitrarily close to the optimal Ω^*.

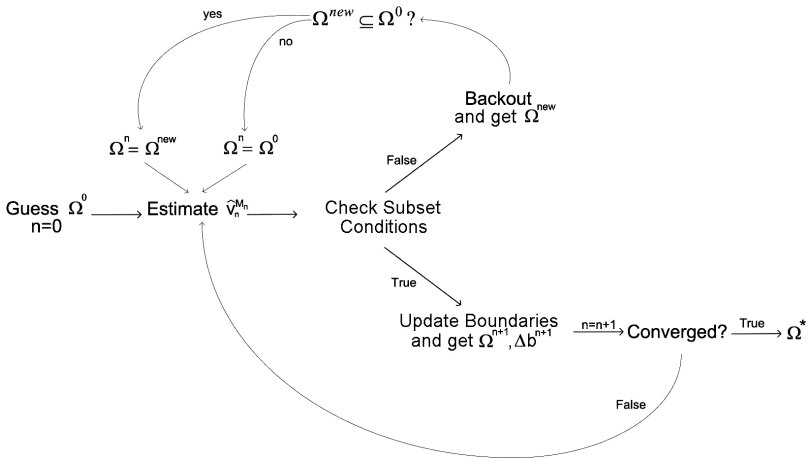

Fig. 6.2 The computational scheme flow.

Though the computational scheme would work and converge for any increasing sequences M and any $a \in (0,1)$, the runtimes of the scheme can greatly be reduced by prudently choosing these, like in many other simulation based schemes like simulated annealing and retrospective approximations. We have found that the following parameter choices perform well and we also use these for all our computational results in Section 6.4: $M_n = M_0 n^{\frac{1}{2}}$ with $M_0 = 200$ and $a = 0.3$.

Table 6.1 Runtime data under two schemes.

N	PDE	Simulation
1	34 sec	4 mins
2	20 mins	58 mins
3	45 hrs	3.4 hrs
4	NA	8.6 hrs
5	NA	18.7 hrs
6	NA	36.6 hrs
7	NA	62.3 hrs

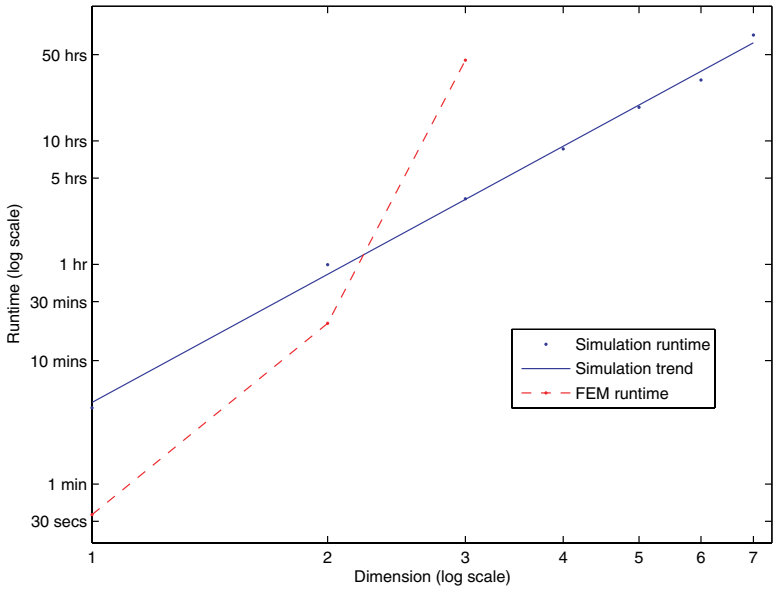

Fig. 6.3 Computational runtime against dimension under two schemes.

6.4 Dimensionality and Scaling

In this section, we first demonstrate that the proposed scheme indeed scales polynomially with dimension and compare it to the runtimes of the scheme that uses FEM to solve for each V_n. The code is implemented in Matlab and the runtimes are based on execution by a single processor Pentium IV machine running at 3 Ghz with 1 GB RAM. We consider a sequence of problems of increasing dimension. The first problem considers only one stock, the second problem considers two stock and so on. For the sake of comparison we always consider independent stock with $\alpha_i = 0.14, \sigma_i = 0.3$, $\lambda_{li} = \lambda_{ui} = 5\%$ and $r = 10\%$. Table 6.1 records the runtimes for each of these problems for both the PDE based scheme and the proposed simulation based scheme. The "NA" (for "Not Available") indicates cases where very large runtimes made computations infeasible.

Figure 6.3 plots the logarithm of the runtimes against logarithm of dimension, showing the nature of runtime scaling. Suppose the runtime t_r is a polynomial function of dimension N then $\log t_r$ and $\log N$ should have a linear relation. Figure 6.3 confirms a linear relation with a slope of 3.4455, implying that $t_r(N) = CN^{3.4455}$ for some constant C.

References

[1] F. AitSahlia and P. Carr, "American options: A comparison of numerical methods," in *Numerical methods in finance*, (L. C. G. Rogers and D. Talay, eds.), pp. 67–87, Cambridge University Press, 1997.

[2] M. Akian, A. Sulem, and M. Taksar, "Dynamic optimization of long-term growth rate for a portfolio with transaction costs and logarithmic utility," *Mathematical Finance*, vol. 11, no. 2, pp. 153–188, 2001.

[3] J. A. Bather, "A Diffusion model for the control of a dam," *Journal of Applied Probability*, vol. 5, pp. 55–71, 1968.

[4] V. Benes, L. Shepp, and H. Witsenhausse, "Some solvable stochastic control problems," *Stochastics*, vol. 4, pp. 39–83, 1980.

[5] A. Bensoussan and J. Lions, *Impulsive Control and Quasi-Variational Inequalities*. Dunod, 1982.

[6] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *Journal of Political Economy*, vol. 81, no. 3, pp. 637–654, 1973.

[7] M. J. Brennan and E. S. Schwartz, "The valuation of American put options," *Journal of Finance*, vol. 32, no. 2, pp. 449–462, 1977.

[8] M. Broadie and J. Detemple, "American option valuation: Approximations, and a comparison of existing methods," *Review of Financial Studies*, vol. 9, no. 4, pp. 1211–1250, 1996.

[9] M. Broadie and J. Detemple, "Option pricing: Valuation models and applications," *Management Science*, vol. 50, no. 9, pp. 1145–1177, 2004.

[10] M. Broadie and P. Glasserman, "Pricing American-style securities by simulation," *Journal of Economic Dynamics and Control*, vol. 21, pp. 1323–1352, 1997.

[11] R. Caldentey and L. Wein, "Revenue management of a make-to-stock queue," *Manufacturing and Service Operations Management*, vol. 4, no. 1, 2002.
[12] P. Carr, R. Jarrow, and R. Myneni, "Alternative characterizations of American put options," *Mathematical Finance*, vol. 2, pp. 87–106, 1992.
[13] G. R. Courtadon, "A more accurate finite difference approximation for the valuation of options," *Journal of Financial and Quantitative Analysis*, vol. 17, pp. 697–703, 1982.
[14] J. C. Cox, S. A. Ross, and M. Rubinstein, "Option pricing: A simplified approach," *Journal of Financial Economics*, vol. 7, pp. 229–263, 1979.
[15] M. Davis and A. Norman, "Portfolio selection with transaction costs," *Mathematics of Operations Research*, vol. 15, pp. 676–713, 1990.
[16] A. Dixit and R. Pindyck, *Investment Under Uncertainty*. Princeton University Press, 1993.
[17] D. Duffie, *Dynamic Asset Pricing Theory*. Princeton University Press, Third ed., 2001.
[18] H. Feng and K. Muthuraman, "A computational method for stochastic impulse control problems," Working Paper, 2008.
[19] H. Feng, K. Muthuraman, and V. Deshpande, "Replensihment policies for multi-product stochastic inventory systems with correlated demand and joint-replenishment costs," Working Paper, 2008.
[20] W. H. Fleming and H. M. Soner, *Controlled Markov Processes and Viscosity Solutions*. Springer, 1993.
[21] A. Friedman, *Variational Principles and Free-Boundary Problems*. Wiley, New York, 1982.
[22] J. M. Harrison, "Brownian models of queueing networks with heterogeneous customer populations," *Stochastic Differential Systems: Stochastic Control Theory and Their Applications,* IMA, vol. 10, pp. 147–186, 1988.
[23] J. Huang, M. Subrahmanyam, and G. Yu, "Pricing and hedging American options: A recursive integration method," *Review of Financial Studies*, vol. 9, pp. 277–300, 1996.
[24] T. J. Hughes, *Finite Element Method: Linear Static and Dynamic Analysis*. Prentice Hall, 1987.
[25] S. D. Jacka, "Optimal stopping and the American put," *Mathematical Finance*, vol. 1, pp. 1–14, 1991.
[26] N. Ju, "Pricing an American option by approximating its early exercise boundary as a multipiece exponential function," *Review of Financial Studies*, vol. 11, no. 3, pp. 627–646, 1998.
[27] I. Karatzas and S. E. Shreve, *Methods of Mathematical Finance*. Springer-Verlag, 1998.
[28] N. Keohane, N. V. Roy, and R. J. Zeckhauser, "Controlling stocks and flows to promote quality: The environment, with applications to physical and human capital," NBER Working Paper, 2000.
[29] I. Kim, "The analytic valuation of American options," *Review of Financial Studies*, vol. 3, pp. 547–572, 1990.
[30] S. Kumar and K. Muthuraman, "A numerical method for solving stochastic singular control problems," *Operations Research*, vol. 52, no. 4, pp. 563–582, 2004.

[31] H. J. Kushner and P. G. Dupuis, *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer, 1992.

[32] H. J. Kushner and L. F. Martins, "Numerical methods for singular stochastic control problems," *SIAM Journal of Control and Optimization*, vol. 29, no. 6, pp. 1443–1475, 1991.

[33] H. G. Landau, "Heat conduction in a melting solid," *Quarterly Applied Mathematics*, vol. 8, p. 81, 1950.

[34] F. A. Longstaff and E. S. Schwartz, "Valuing American options by simulation: Simple least-squares approach," *Review of Financial Studies*, vol. 14, pp. 113–147, 2001.

[35] D. Luenberger, *Investement Science*. Oxford University Press, 1998.

[36] M. J. P. Magill and G. M. Constantinides, "Portfolio selection with transaction costs," *Journal of Economic Theory*, vol. 13, pp. 245–263, 1976.

[37] R. McDonald and M. Schroder, "A parity result for American options," *Journal of Computational Finance*, vol. 1, pp. 5–13, 1998.

[38] H. P. McKean, "Appendix: A free boundary problem for the heat equation arising from a problem in mathematical economics," *Industrial Management Review*, vol. 6, pp. 32–39, 1965.

[39] R. C. Merton, "Lifetime portfolio selection under uncertainty: The continuous time case," *The Review of Economics and Statistics*, vol. 51, pp. 247–257, 1969.

[40] P. L. J. V. Moerbeke, "On optimal stopping and free boundary problems," *Archive for Rational Mechanics and Analysis*, vol. 60, pp. 101–148, 1976.

[41] K. Muthuraman, "A computational scheme for optimal investment-consumption with proportional transaction costs," *Journal of Economic Dynamics and Control*, vol. 31, no. 4, pp. 1132–1159, 2007.

[42] K. Muthuraman, "A moving boundary approach to American option pricing," *Journal of Economic Dynamics and Control*, 2008.

[43] K. Muthuraman and S. Kumar, "Multi-dimensional portfolio optimization with proportional transaction costs," *Mathematical Finance*, vol. 16, no. 2, pp. 301–335, 2006.

[44] K. Muthuraman and H. Zha, "Simulation based portfolio optimization for large portfolios with transaction costs," *Mathematical Finance*, vol. 18, no. 1, pp. 115–134, 2008.

[45] R. Myneni, "The pricing of the American option," *The Annals of Applied Probability*, vol. 2, no. 1, pp. 1–23, 1992.

[46] B. F. Nielsen, O. Skavhaug, and A. Tveito, "Penalty and front-fixing methods for the numerical solution of American option problems," *The Journal of Computational Finance*, vol. 5, no. 4, pp. 69–97, 2002.

[47] J. T. Oden and J. N. Reddy, *An Introduction to the Mathematical Theory of Finite Elements*. Academic Press, 1978.

[48] E. Schwartz, "The valuation of warrants: Implementing a new approach," *Journal of Financial Economics*, vol. 4, pp. 79–93, 1977.

[49] S. E. Shreve and H. M. Soner, "Optimal investment and consumption with transaction costs," *Annals of Applied Probability*, vol. 4, no. 3, pp. 609–692, 1994.

[50] M. Taksar, M. J. Klass, and D. Assaf, "A diffusion model for optimal portfolio selection in the presence of Brokerage fees," *Mathematics of Operations Research*, vol. 13, pp. 277–294, 1988.

[51] J. A. Tilley, "Valuing American options in a path simulation model," *Transactions of the Society of Actuaries*, vol. 45, pp. 83–104, 1993.

[52] L. Wu and Y. Kwok, "A front-fixing finite difference method for the valuation of American options," *Journal of Financial Engineering*, vol. 6, pp. 83–97, 1997.

[53] T. Zariphopolou, "Transaction costs in portfolio management and derivative pricing," *Proceedings of Symposia in Applied Mathematics*, vol. 57, pp. 101–163, 1999.