Quantitative Methods for Investment Decisions in Communication Networks

Clare Saddler

Thesis submitted for the degree of

Doctor of Philosophy

in

Applied Mathematics

at

The University of Adelaide

(Faculty of Mathematical and Computer Sciences)

Department of Applied Mathematics

September 2006

Contents

List of Tables

List of Figures

Abstract

Information and communication technology (ICT) is a multi-billion dollar industry [11]. It is therefore paramount to employ the most up to date decision making strategies when making ICT investments. This thesis uses a framework, originally developed for perpetual American call options, to study pertinent issues in this industry.

The models in existing literature often assume that investment costs are fixed, but in the ICT industry we expect the cost to decrease exponentially according to Moore's and Gilders' laws. The models are therefore extended to support decreasing costs. The investment values and stopping times are determined for various decay parameters. For large decay parameters, we find that the investment values are close in geometric Brownian motion (GBM) and multiplicative jump-diffusion process (JDP) models. Typical error scenarios are explored and the models are found to be fairly robust.

Once a network link has been built, its capacity can be increased by upgrading hardware in the associated switches. We initially develop a general strategy for deciding when to make this investment and find an analytical solution for a GBM demand process. A logistic process is then used to model demand with saturation and Kummer's equation is used to find an analytical solution for the increasing capacity model. In the GBM model, there is a unique optimal trigger which is greater than the link's initial capacity. Compared to the GBM model, investments are made later in the demand saturation model and yield lower investment values. Furthermore, in some extreme cases, we find that the optimal trigger does not exist.

Declaration

This work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text.

I give consent to this copy of my thesis, when deposited in the University Library, being made available in all forms of media, now or hereafter known.

Clare Saddler

September 25, 2006

Acknowledgements

I am indebted to John van der Hoek for his guidance and support throughout my program. He also provided invaluable assistance with the models and expanded my knowledge through seminars and suggested readings. I am also grateful to Matthew Roughan for suggesting several ICT applications and providing general advice. I also appreciate the general assistance which I received from Liz Cousins, Dianne Parish, Peter Gill and Charles Pearce.

I am especially grateful to my partner Sven Delaney for his encouragement and companionship. I am also grateful to Kathleen Lumley College for providing many new friends, in particular Kwoping Tam, Sarah Meehan, Anand Prabhu, David Thorn and James Menzies. I am also grateful to Mark O'Donoghue and Greg Smith for providing meditation and yoga classes.

Chapter 1

Introduction

The information communications technology (ICT) industry is a multi-billion dollar industry. In 2005, the Australian telecom market alone was estimated to be worth \$31 billion [11]. Given the magnitude of this market, decision making processes which yield optimal strategies may increase profits by millions. This thesis determines whether or when an investor should make ICT investments. Similar problems have been studied in the manufacturing industry [22, 43], but these models fail to address pertinent issues in the ICT industry.

The investment problem depends on two functions: the current value of future revenues $V(t)$ (which we shall henceforth refer to as the *value process*), and the investment cost $I(t)$. Traditional Investment Analysis is based Net Present Value (NPV) analysis. The NPV is simply the current value of future revenues less the current cost.

$$NPV = V(0) - I(0).$$

If this value is positive the investment is made immediately. Otherwise the investment is never made. This method is often ignored in practice because it does not allow for management flexibility. For instance, in the current problem it ignores the fact that the decision can be postponed.

Decision Analysis provides an alternative approach. Figure 1.1 shows a simple

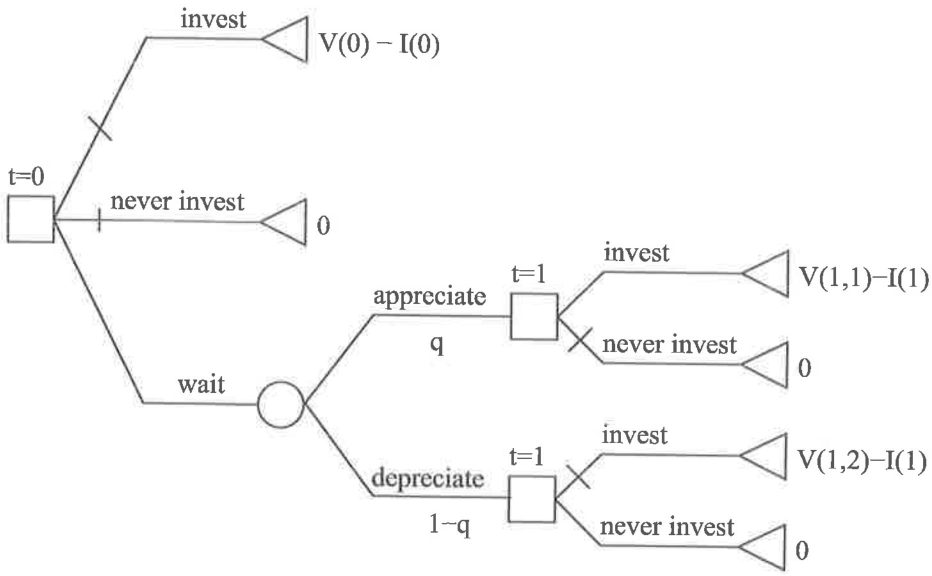


Figure 1.1: A simple decision tree

decision tree which includes the two choices given in the NPV model: invest immediately or never invest. It also includes a third option: wait until time t=1, then apply the NPV rule to decide whether to invest or not. In this simple model the value process is expected to appreciate or depreciate with risk-neutral probabilities q and $1-q$ respectively. The decision tree is solved by evaluating options at each branch. At time t=1, there are two possible scenarios: either the value process has appreciated to $V(1,1)$ or it has depreciated to $V(1,2)$. Applying the NPV rule, the value of each investment is $(V(1,1)-I(1))^+$ and $(V(1,2)-I(1))^+$ respectively. The value of waiting is the the expected value of these two values and discounted by e^r, where r is the risk-free interest rate for the period to $t=1$,

$$W(0)=\frac{q(V(1,1)-I(1))^+ + (1-q)(V(1,2)-I(1))^+}{e^r}.$$

This value is then compared with the value of investing immediately $V(0)-I$ and the value of never investing 0. The optimal strategy is to select the decision which yields the greatest value.

Decision analysis has greater flexibility than NPV analysis alone, but it requires considerable time to calculate the decision trees, and decision trees only consider time points that have been enumerated. For instance this decision tree will determine whether it is better to invest at $t = 0$ or $t = 1$, but it ignores other choices which may yield a greater profit (e.g. $t = 0.5$).

This leads to a new approach called real options which is based on option pricing techniques originally developed for financial options. Real options acknowledges management flexibility, including the ability to postpone investments. Several models have been developed for the deferred investment problem [22, 41, 43, 32]. However, most of these models were developed for manufacturing investments. This thesis uses real options to study ICT investments. But before discussing the work in this thesis, some background information on financial options and real options is needed.

1.1 Financial Options

Financial options can be used to provide protection against fluctuations in the market and they also enable investors to increase their potential gains (and losses) for a limited amount of capital. A *call stock option* gives the owner the right to buy a stock S for a given price K (known as the *strike price*). In contrast, a *put stock option* gives the owner the right to sell the stock for the strike price. *European* options can only be exercised at the expiry date T. *American* options can be exercised at anytime up to the expiry date. At the time of exercise the owner receives a payoff. The payoff on a call option is $(S(t) - K)^+ = \max(0, S(t) - K)$ and the payoff on a put option is $(K - S(t))^+ = \max(0, K - S(t))$, where $S(t)$ is the stock price at time t. Since the seller is obliged to cover the cost for all possible circumstances we need to find a fair price for the option.

Option pricing models are based on setting up a portfolio of a risk-free asset (e.g. a bond) and a risky asset $S(t)$ (i.e. the underlying stock). The stock price $S(t)$ may typically be assumed to follow a geometric Brownian motion (GBM)

$$dS(t) = \nu S(t)dt + \sigma S(t)dB(t), \tag{1.1}$$

where ν is the drift, σ is the volatility and $B(t)$ is a standard Brownian motion. In 1973, Black and Scholes [7] and Merton [44] derived a partial differential equation for the call option value

$$\frac{\partial C}{\partial t} - rC + rS\frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} = 0. \tag{1.2}$$

Black and Scholes found a closed form solution for European call options with no dividends, and Merton generalized this result for European options with continuous dividends. The Black-Scholes formula for a European call option with strike price K and dividend rate δ is

$$C(S,t) = Se^{-\delta(T-t)}N(d_1) - Ke^{-r(T-t)}N(d_2), \tag{1.3}$$

where

$$\begin{aligned} d_1 &= \frac{\log(S/K) + (r - \delta + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}, \\ d_2 &= d_1 - \sigma\sqrt{T-t}, \\ N(x) &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} \exp\left(-\frac{z^2}{2}\right) dz. \end{aligned}$$

The Black and Scholes and Merton formula [7, 44] gives an exact solution for European call options (see Figure 1.2). If an American call option has no dividends (i.e. $\delta = 0$), it is better to wait until the last possible moment (i.e. the expiry date). In this case the American call option value is equal to the European call option value. The American call option with dividends cannot be solved analytically and must be estimated using approximation techniques (e.g. binomial models [17, 52], finite-difference models [9, 33, 64] or simulation techniques [8, 10]).

Perpetual American options, which have no expiry date, may be used for options with large expiry dates. In this thesis, we use perpetual American call options to value investment decisions. Perpetual American options are useful because they

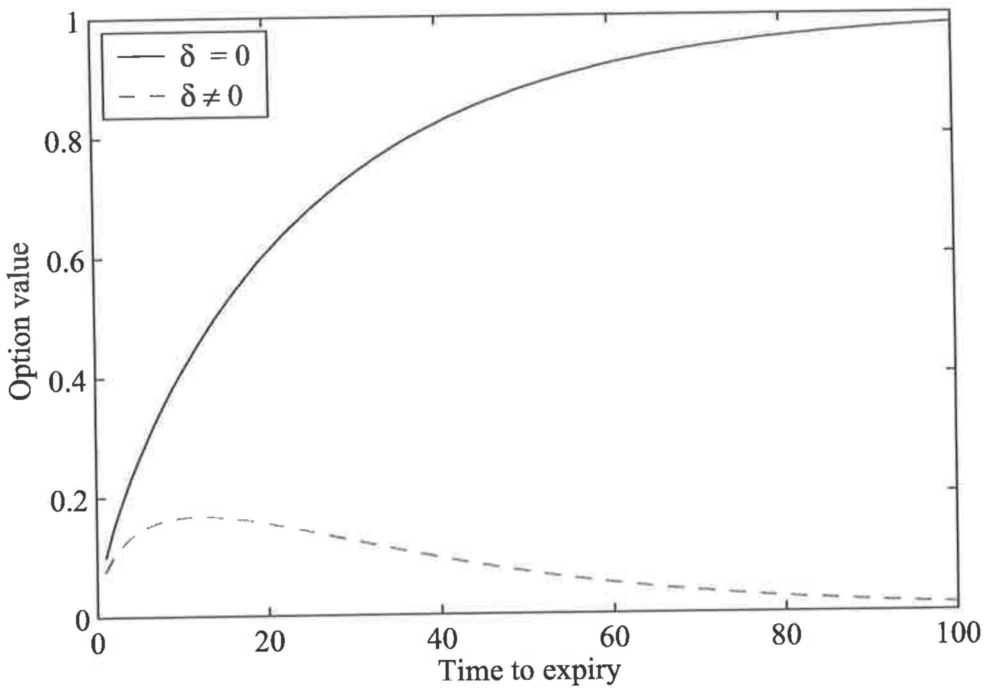


Figure 1.2: The Black-Scholes values for European call options

often have closed form prices and are therefore more tractable than standard American options [35, 25]. They also specify optimal strategies for exercising options. Since the owner does not know the future behaviour of the stock price, the exercise strategy can only depend on the current state of this underlying process and its history. Fortunately, the optimal strategy only depends on the current state; the optimal strategy tells the owner to exercise the option when the stock price $S(t)$ first reaches some threshold S^* (known as the *option trigger*). Since the perpetual model assumes large expiry dates, we use the binomial model to determine whether the perpetual model is suitable for typical expiry dates. Technical details of the perpetual model and the binomial model are provided in Section 2.3.

1.2 Real Options

Real Options Analysis (ROA) is a new approach to capital budgeting that uses option pricing techniques to value real investments. Unlike traditional capital budgeting approaches, ROA is able to incorporate multiple outcomes and management flexibility. Trigeorgis [60] describes the following options in his book *Real Options*:

- The option to defer an investment.
- The option to abandon a project.
- The option to switch inputs.
- The option to switch outputs.
- The option to stage investments.
- The option to alter operating scale.

Much of the real options literature assumes that the underlying asset is tradeable or synthetically so by what is called what is often called a twin security (see [61, 22]). This enables one to apply the option pricing techniques directly. Some new methods have been developed to support derivatives on non-tradeable assets [32, 24, 47]. Most of this work is concerned with finite-time models, although Henderson [32] considers a perpetual investment model. This approach is beyond the scope of this thesis but may be applied in future work.

Under the twin-security assumption, the investment problem is equivalent to a call option where the investment cost $I(t)$ is the strike price and the present value of future revenues $V(t)$ replaces the stock price. Bhagat [5] suggested that the European call model be used for *now or never* investment decisions. But since most investment decisions can be postponed, it is usually more appropriate to use American call options. The investor can then enter the project at any time t (and thereby gain the right to any future revenues) for the investment cost, resulting in a payoff of $(V(t) - I(t))^+$.

As discussed in Section 1.1, the Black-Scholes formula does not apply to American call options on dividend paying stock. While some finite-time models (e.g. the binomial model) have been used to study investment decisions [61], the perpetual model is frequently used for investments with large durations [43, 22, 41]. McDonald and Siegal first used perpetual models to study the investment decision. Dixit and Pindyck considered many variations of this model in their seminal book on real options [22]. In the simplest case, the value process $V(t)$ is modelled using a geometric Brownian Motion. They also developed a jump-diffusion model which has subsequently been used to model bandwidth investments [41]. In this thesis, we shall explore both models before developing specific models for ICT investments.

Most real option models assume that the investor has monopoly. However, a new branch of real options which incorporates game theory, called *option games*, has been used to study investment problems when there is some competition in the market [28]. This approach is beyond the scope of the present work but may be applied in future work.

Real options have been used in a wide range of areas including capital budgeting [57], natural resources [59], manufacturing [39], foreign investment [4], research and development [46], shipping [6] and nuclear waste management [12]. Real options have also been used to make investments in optical networks [41, 20, 36] and wireless networks [19]. This thesis is concerned with two investment decisions in optical networks:

- the option to build new infrastructure (e.g. add a new link),
- the option to increase capacity on an existing link.

In optical networks, a link's transmission capacity can be increased by upgrading hardware (i.e. the switching cards) in the switches at either end of the link. Table 1.1 lists some common optical carrier (OC) levels [63]. Lassila [41] used a jump-diffusion process (JDP) with upward jumps to model bandwidth supply, and showed that the value process will be a JDP with downward jumps when the de-

Rate	Max Capacity (Mbps)
OC-1	52
OC-12	622
OC-48	2488
OC-192	9953

Table 1.1: Optical Carrier (OC) Levels

mand process is a GBM and the supply process is a JDP with upward jumps. For simplicity, we shall henceforth we shall refer to JDPs with upward jumps and downward jumps as positive JDPs and negative JDPs respectively. The decision to use positive jump-diffusion process to model bandwidth supply is based on the fact that supply increases dramatically when a big player makes an investment (e.g. cable laid down in the pacific ocean between Sydney and Los Angeles in the late 90s, and switches upgraded from OC-48 to OC-192 on cable running in conduits in the Rocky Mountains). D'Halluin, Forsyth and Vetzal [20] developed some Partial Differential Equations (PDEs) for increasing capacity on existing links and used a numerical PDE solver to find the optimal solution. In this thesis, we shall extend Lassila's model and develop two analytical models for increasing link capacity.

Real options have also been applied to the access pricing debate. Access pricing issues arise in many industries that were formally considered natural monopolies (e.g. telecom, postal services, electricity, gas and railways). In the past, governments licensed a single provider for telecom services because it was considered inefficient to duplicate the network architecture. In the 1980s, this approach was criticized because the service provider could finance inefficient practices by simply raising the cost to customers, and new providers were allowed to enter the market. Since entrants require access to the existing network they need to pay a tariff to the incumbent. In most countries, these access prices are regulated by the government [40]. Hausman [30, 31] claimed that existing policies gave entrants *real options* for free and used an example from Dixit and Pindyck [22] to justify higher access prices. Two simple models for access pricing models have also been proposed [14, 58] but many authors have questioned the application of real options to access pricing [2, 49, 23, 13]. Access pricing issues will not be explicitly addressed in this thesis but some of the work in this thesis could be applied to the access pricing debate.

1.3 Description of the Project

This thesis uses perpetual American call models to study two investment decisions in optical networks:

- the option to build new infrastructure (e.g. add a new link),
- the option to increase capacity on an existing link (e.g. OC-48 to OC-192).

We begin with two existing fixed-cost models for building new infrastructure in which the value process is assumed to follow either a geometric Brownian motion (GBM) or a multiplicative jump-diffusion process (JDP). These models were originally solved using a PDE approach. We provide an alternative derivation using martingale methods and investigate some timing issues that were not addressed in the previous literature. These models are then extended to support decreasing investment costs according to Moore's and Gilder's laws and some common error scenarios are investigated.

As mentioned in Section 1.2, D'Halluin et al. [20] developed a PDE for increasing link capacity and used a numerical PDE solver to find an optimal solution. We develop a similar PDE for increasing link capacity and find an analytical solution for a GBM demand process. For finite populations we expect the demand to level off, and so another analytical solution is developed for a logistic demand process. We then compare the results for the GBM and logistic models.

1.4 Thesis Outline

Chapter 2 (Background) provides some background material for the thesis. First, we present some key concepts from stochastic calculus which are used in the thesis: standard Brownian motions, Poisson processes, Itô's lemma, stopping times, and the optional sampling theorem. Next, we describe three stochastic processes used in the thesis:

- A geometric Brownian motion (GBM), defined by

$$dY(t) = \nu Y(t)dt + \sigma Y(t)dB(t),$$

  where ν is the drift, σ is the volatility and $B(t)$ is a standard Brownian motion.

- A multiplicative jump-diffusion process (JDP), defined by

$$dY(t) = \nu Y(t)dt + \sigma Y(t)dB(t) + \phi Y(t)dN(t),$$

  where ν is the drift, σ is the volatility, ϕ is the jump magnitude, $B(t)$ is a standard Brownian motion, and $N(t)$ is a Poisson process with arrival rate λ.

- A logistic process (LP), defined by

$$dY(t) = \eta(\overline{Y} - Y(t))Y(t)dt + \sigma Y(t)dB(t),$$

  where η is the speed of reversion, $\overline{Y}$ is the long-run equilibrium level, σ is the volatility and $B(t)$ is a standard Brownian motion.

Finally, we provide some technical details for the perpetual and binomial models.

Chapter 3 (Building New Infrastructure) considers two existing models for building new infrastructure: a GBM model and a JDP model. In each case, the optimal risk-neutral expected investment value is given by

$$F(V) = \max_{\tau \geq 0} E\left[(V(\tau) - I(\tau))^{+} e^{-r\tau}\right],$$

where $V(t)$ is the time t value of future revenues (with $V(0) = V$), $I(t) \equiv I$ is the cost of investing at time t, τ is a stopping time, and r is the risk-free rate. Both models were originally solved using a Partial Differential Equation (PDE) approach. We provide an alternative derivation using martingale techniques. We then apply a binomial model to two examples from the literature to determine whether the perpetual model is appropriate for typical expiry dates. In one of these examples, the perpetual model is only accurate for expiry dates greater than 60 years. Finally, we use stopping times to determine investment times. Simulation programs are used to estimate the investment times for both processes. We also used an analytic solution to obtain more precise values for the GBM model, and then compared these values with the simulation results.

Chapter 4 (Decreasing Investment Costs) considers investments with decreasing costs. A negative exponential cost function

$$I(t) = Ie^{-\alpha t},$$

with a decay parameter $\alpha \geq 0$, is used to support decreasing costs. We observe similar behaviour in the GBM and JDP models, and show that investment values in a JDP model converge to those in a related GBM model as the decay parameter increases. We then test the robustness of the model by considering various error scenarios. The model is found to be robust. Relatively large errors do not reduce the investment value by more than 5% (although extremely large errors may cause a loss of 30% or even result in negative payoffs in special cases). The results demonstrate the value of following the optimal strategy.

Chapter 5 (Increasing Link Capacity) considers the option of increasing link capacity. If the capacity is increased from S_0 to S_1 when the demand process $D(t)$ (with $D(0) = D$) reaches the threshold y, then the risk-neutral expected investment value is

$$F(D, y) = E\left[\int_0^\tau \beta \min(D(t), S_0)e^{-rt}dt + \int_\tau^\infty \beta \min(D(t), S_1)e^{-rt}dt - Ie^{-(r+\alpha)\tau}\right],$$

where

$$\tau = \inf\{t \geq 0 : D(t) \geq y\},$$

β is revenue per connection, r is the risk-free rate, I is the initial investment cost and α is the decay parameter. We seek optimal trigger

$$y^* = \underset{y}{\operatorname{argmax}} F(D, y).$$

We consider a GBM demand process

$$dD(t) = \nu D(t)dt + \sigma D(t)dB(t),$$

and find an analytical solution using a PDE approach. We find that the optimal trigger is never greater than the initial capacity S_0, and prove that the optimal trigger exists and is unique. We also provide a method for finding the optimal trigger and perform some numerical analysis.

Chapter 6 (Demand Saturation) considers a logistic demand process

$$dD(t) = \eta(\overline{D} - D(t))D(t)dt + \sigma D(t)dB(t).$$

We find an analytical solution using a PDE approach with Kummer's equation. To compare the GBM and LP models, we define a logistic variation of the GBM, by choosing $\overline{D}$ and then setting $\eta = \frac{\nu}{\overline{D}}$. As $\overline{D} \to \infty$, the logistic process converges to the original GBM and so we can treat the GBM as a special case of the LP with $\overline{D} = \infty$. We find that reducing $\overline{D}$ leads to later investment times and smaller investment values. For extremely large investment costs, we find that the optimal trigger may not exist in some logistic models. In this case it is always better to wait and so the investment will never be made.

Chapter 7 (Summary and Conclusions) presents some conclusions and describes possible future directions from the work described in this thesis.

Appendices A-D provide some additional material that could not be included in the main body of the thesis: some pertinent mathematical theory and results, Java programs and Matlab files.

Chapter 2

Background

This chapter provides some background material for the thesis: some key concepts of stochastic calculus, a description of three stochastic processes and some technical details for the perpetual model and the binomial model.

2.1 Stochastic Calculus

In this section we provide some key concepts of stochastic calculus that will be used in this thesis: Brownian motions, Poisson processes, Itô's lemma, stopping times and the optional sampling theorem.

2.1.1 Brownian motions and Poisson processes

Two adapted processes are used to model random phenomena: the standard Brownian motion $B(t)$ and the Poisson process $N(t)$. The definitions for adapted processes, the standard Brownian motion (also known as the Weiner process) and the Poisson process are given below.

Definition 2.1 (Adapted Processes[18]). We say that a process $(X_t, t \geq 0)$ is $\mathcal{F}_t$-adapted if X_t is $\mathcal{F}_t$-measurable for any t.

Definition 2.2 (Brownian Motion [34, page 47]). A *(standard, one-dimensional) Brownian motion* is a continous, adapted process $B = \{B_t, \mathcal{F}_t; 0 \leq t < \infty\}$, defined on some probability space $(\Omega, \mathcal{F}, P)$, with the properties:

1. $B_0 = 0$ a.s.,
2. for $0 \leq s < t$, $B_t - B_s$ is independent of $\mathcal{F}_s$, and
3. $B_t - B_s$ is normally distributed with mean 0 and variance $t - s$.

Definition 2.3 (Poisson Process [34, page 12]). A *Poisson process with intensity* $\lambda > 0$ is an adapted, non-negative integer-valued, right-continous with left limits (RCLL) process $N = \{N_t, \mathcal{F}_t; 0 \leq t < \infty\}$ such that $N_0 = 0$ almost surely, and for $0 \leq s < t$, $N_t - N_s$ is independent of $\mathcal{F}_s$ and is Poisson distributed with mean $\lambda(t - s)$.

2.1.2 Itô's Lemma

Protter [50]'s version of Itô's lemma is needed to handle jumps. The definitions for martingales and semimartingales and Itô's lemma are given below [34, 50].

Definition 2.4. The process $\{X_t, \mathcal{F}_t; 0 \leq t < \infty\}$ is said to be a *submartingale* (respectively, a *supermartingale*) if we have $E[X_t|\mathcal{F}_s] \geq X_s$ (respectively, $E[X_t|\mathcal{F}_s] \leq X_s$) P-a.s., for every $0 \leq s < t < \infty$. We shall say that $\{X_t, \mathcal{F}_t; 0 \leq t < \infty\}$ is a *martingale* if it is both a submartingale and a supermartingale.

Definition 2.5. A *continuous semimartingale* $X = \{X_t, \mathcal{F}_t; 0 \leq t < \infty\}$ is an adapted process which has the decomposition, almost surely with probability P (P-a.s.),

$$X_t = X_0 + M_t + A_t;\ 0 \leq t < \infty, \tag{2.1}$$

where X_0 is $\mathcal{F}_t$-measurable, $M_0 = A_0 = 0$, $M = \{M_t, \mathcal{F}_t; 0 \leq t < \infty\}$ is a continuous local martingale and $A = \{A_t, \mathcal{F}_t; 0 \leq t < \infty\}$ is continuous adapted process of finite variation.

Theorem 2.1 (Itô's lemma [50, page 71]). *Let X be a semimartingale and let f be a C^2 real function. Then $f(X)$ is again a semimartingale, and the following formula holds:*

$$\begin{aligned} f(X_t) - f(X_0) &= \int_{0+}^{t} f'(X_{s-})dX_s + \frac{1}{2}\int_{0+}^{t} f''(X_{s-})d[X,X]_s^c \\ &+ \sum_{0<s\leq t} \{f(X_s) - f(X_{s-}) - f'(X_{s-})\Delta X_s\}. \end{aligned}$$

Proof. See Protter [50, pages 71–74]. □

2.1.3 Stopping Times

Stopping times are used to measure the first time that a stochastic process $Y(t)$ satisfies a pre-defined stopping rule. Clearly, the stopping rule can only depend on the current state or history of the process. For example, a stopping rule in the gamblers ruin problem could be when the gambler runs out of money. In general, a stopping time is a map $\tau : \Omega \longrightarrow [0, \infty)$ so that for each $t \geq 0$, the set $\{\omega \in \Omega | \tau(\omega) \leq t\}$ is $\mathcal{F}_t$-measurable, where $\mathcal{F}_t = \sigma\{Y(s)|s \leq t\}$. This means that the stopping time τ has the property that $\tau \leq t$ is determined by the values of Y up to and including time t. We note that some of the stopping times in this thesis are not almost surely finite (i.e. $P(\tau < \infty) < 1$). In option pricing theory, stopping times are used to measure the time when the option is exercised. The optimal strategy, for the perpetual American call model, is to invest when the the stock price $S(t)$ reaches some threshold S^* and so the stopping time is

$$\tau = \{t \geq 0 : S(t) = S^*\}.$$

If the underlying stochastic process (e.g. $Y(t)$ or $S(t)$) is a geometric Brownian motion we can re-write the stopping time in terms of a Brownian motion $X(t)$ and thereby utilize Harrison's formula below,

$$\tau_m = \inf\{t > 0; X(t) \geq m\},$$

where

$$X(t) = \mu t + B(t).$$

Harrison [29] derived the following expression for the distribution of stopping times:

$$P(\tau_m > t) = N\left(\frac{m - \mu t}{\sqrt{t}}\right) - e^{2\mu m} N\left(\frac{-m - \mu t}{\sqrt{t}}\right), \tag{2.2}$$

where $N(x)$ is the cumulative normal distribution,

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{z^2}{2}\right) dz.$$

Rearranging (2.2) gives the probability that the stopping time τ_m is less than some time t,

$$\begin{aligned} G(t) &\equiv P(\tau_m \leq t) \\ &= N\left(\frac{\mu t - m}{\sqrt{t}}\right) + e^{2\mu m} N\left(\frac{-m - \mu t}{\sqrt{t}}\right), \end{aligned} \tag{2.3}$$

Furthermore, taking limits in (2.2) as $t \to \infty$, gives the probability that the stopping time is finite

$$P(\tau_m < \infty) = \begin{cases} 1, & \mu \geq 0; \\ e^{2m\mu}, & \mu < 0. \end{cases} \tag{2.4}$$

Thus the stopping time is almost surely finite for non-negative drift (i.e. $\mu \geq 0$). Figures 2.1 and 2.2 show typical sample paths for $\mu > 0$ and $\mu < 0$, respectively.

2.1.4 Optional Sampling Theorem

The optional sampling theorem will be used to study investment models. We note that the martingales in this thesis do not have a final element X_∞ and so we need to define a bounded time $\tau_n = \tau \wedge n$, apply the first case in Theorem 2.2, and then let $n \to \infty$.

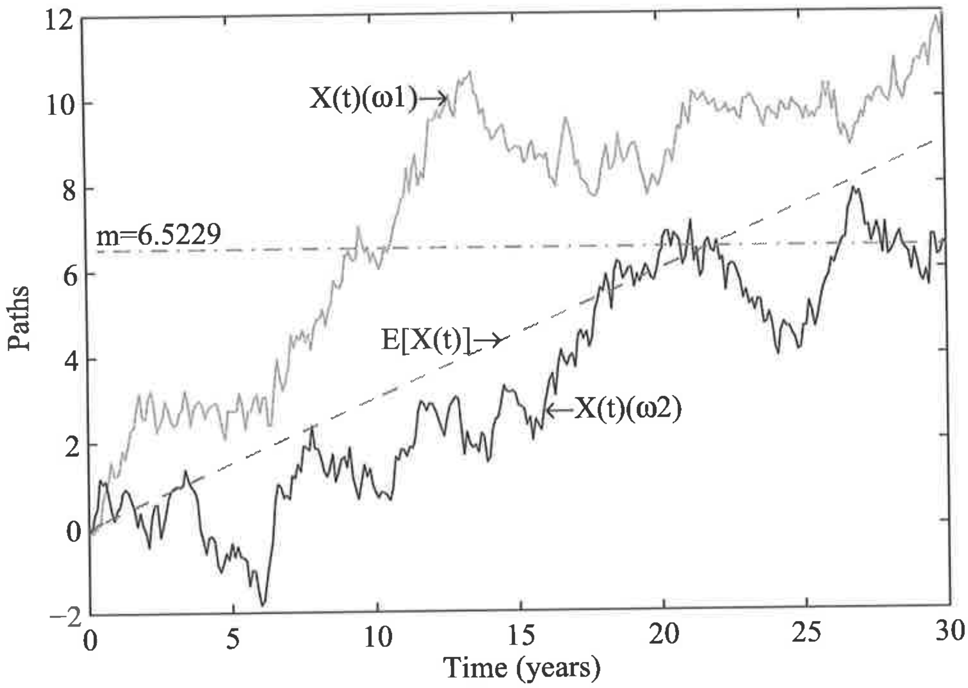


Figure 2.1: Sample paths for $X(t) = 0.3t + B(t)$

Theorem 2.2 (Optional Sampling [53, page 69]). *If X is a martingale and S, T are two bounded stopping times with $S \leq T$,*

$$X_S = E[X_T|\mathcal{F}_S] \text{ almost surely (a.s.)}$$

If X is uniformly integrable, the family $\{X_S\}$ where S runs through the set of all stopping times is uniformly integrable and if $S \leq T$

$$X_S = E[X_T|\mathcal{F}_S] = E[X_\infty|\mathcal{F}_S] \text{ a.s.}$$

Proof. See Revuz and Yor [53, page 69]. □

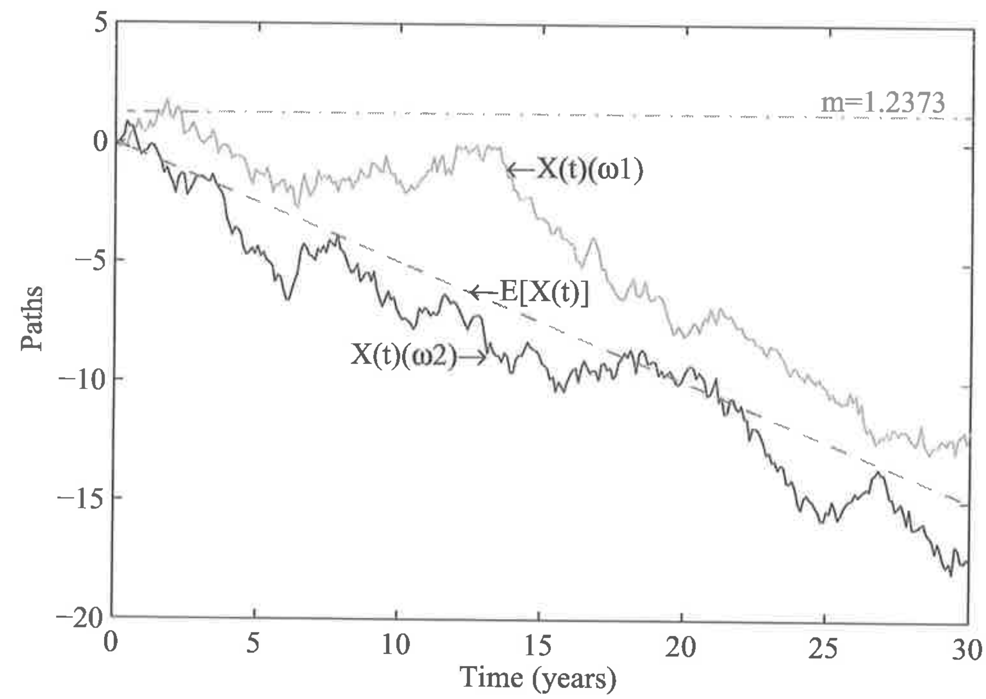


Figure 2.2: Sample paths for $X(t) = -0.5t + B(t)$

2.2 Stochastic Processes

Stochastic processes are used to model stock prices. Louis Bachelier first modelled the stock price using a Brownian motion in 1900 [3]. Unfortunately this model permits negative values (see Figure 2.1). In 1965, Paul Samuelson [55] improved the model by using a geometric Brownian motion, which cannot become negative (see Figure 2.3). Nowadays, geometric Brownian motions are often used to model asset prices and Brownian motions are used to model changes in price. However, other stochastic processes can also be used to model prices [22, 33, 41]. In this section we describe three stochastic processes that will be used in this thesis: geometric Brownian motions, jump-diffusion processes and mean-reversion processes.

2.2.1 Geometric Brownian Motions

Geometric Brownian motions are most commonly used to model stock prices. The geometric Brownian motion, $Y(t)$, is defined by a stochastic differential equation

$$dY(t) = \nu Y(t)dt + \sigma Y(t)dB(t),$$

where ν is the drift, σ is the volatility and $\{B(t) : t \geq 0\}$ is a standard Brownian motion (or Wiener process), and has an exact solution (see Theorem B.1)

$$Y(t) = Y \exp\left\{\left(\nu - \frac{\sigma^2}{2}\right)t + \sigma B(t)\right\}. \tag{2.5}$$

Unless otherwise stated the geometric Brownian motions in this thesis were simulated using the exact solution.

Algorithm 2.3. *To simulate the geometric Brownian motion $Y(t)$ given by (2.5): For $s = 0$ to t by Δ_t*

- *Generate $\epsilon \sim N(0,1)$.*
- *Set $Y(s+\Delta_t) := Y(s)\exp((\nu - \sigma^2/2)\Delta_t + \sigma\epsilon\sqrt{\Delta_t})$.*

Geometric Brownian motions grow or decay exponentially on average according to the the drift term ν. For positive drift (i.e. $\nu > 0$), the process grows exponentially. In fact, the exponential function

$$\overline{Y}(t) = E[Y(t)] = Y\exp\{\nu t\}$$

is the average value for $Y(t)$. Figure 2.3 shows some sample paths for $Y(t) = \exp\{0.06t + 0.2B(t)\}$. The function $\overline{Y}(t) = \exp\{0.08t\}$, represented by the dashed line, gives the expected value of $Y(t)$. The magnitude of the volatility σ determines how far the process will deviate from $\overline{Y}(t)$. In fact,

$$E[(Y(t) - \overline{Y}(t))^2] = Y^2 e^{2\nu t}[e^{\sigma^2 t} - 1].$$

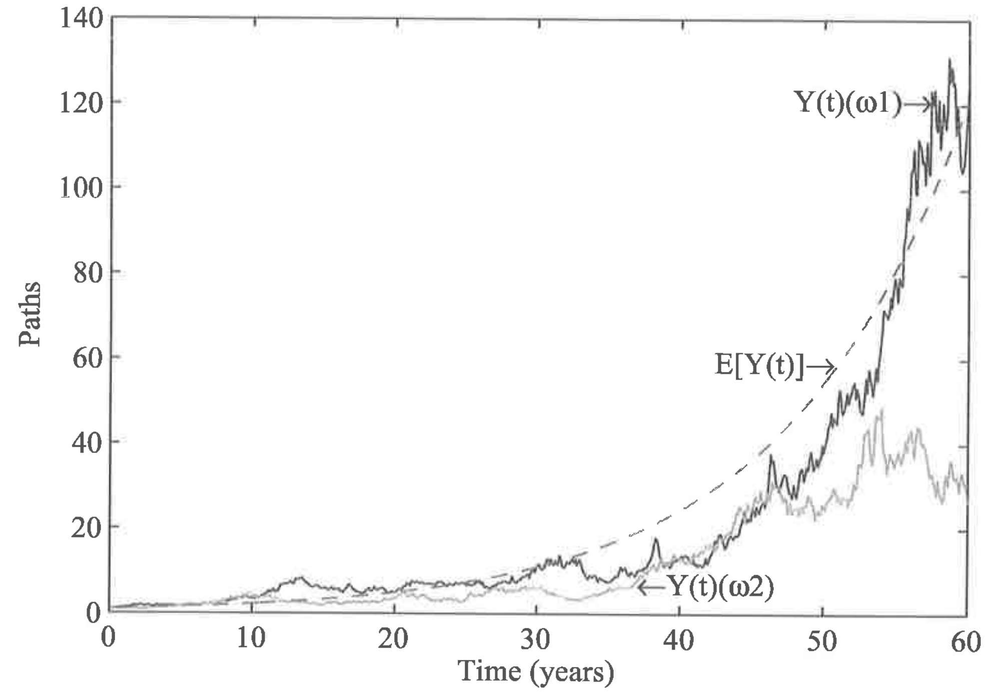


Figure 2.3: Sample paths for a geometric Brownian motion

2.2.2 Jump-Diffusion Processes

Jump-diffusion processes are used to model stock prices that incur discrete jumps (e.g. bandwidth prices [41]). The double exponential jump-diffusion process has the dynamics [38]

$$dY(t) = \nu Y(t-)dt + \sigma Y(t-)dB(t) + Y(t-)d\left(\sum_{i=1}^{N(t)}(V_i - 1)\right). \qquad (2.6)$$

This stochastic differential equation has two random components: the Brownian motion $B(t)$ and the Poisson process $N(t)$. The Poisson process $N(t)$ has an arrival rate λ. The sequence $\{V_i\}$ are i.i.d nonnegative random variables with $Z = \log(V)$ having density

$$f_Z(z) = p \cdot \eta_1 e^{-\eta_1 z} 1_{\{z \geq 0\}} + (1-p) \cdot \eta_2 e^{\eta_2 z} 1_{\{z<0\}}, \; \eta_1 > 1, \; \eta_2 > 1.$$

Figure 2.4 shows some sample paths for this process. Since the jump size is large relative to the volatility, it is not difficult to discern the downward jumps in each sample path. The process is simulated using the return process $X(t) := \log\left(\frac{Y(t)}{Y(0)}\right)$, which is given by

$$X(t) = \left(\nu - \frac{\sigma^2}{2}\right)t + \sigma B(t) + \sum_{i=1}^{N(t)} Y_i, \ X(0) = 0.$$

Choosing the step-size Δ_t sufficiently small, we can assume that at most one jump will occur in the interval $(s, s+\Delta_t]$. Uniform variates ($U_i \sim U(0,1)$) [42] are used to determine whether one or zero jumps will occur and the direction of the jump (up or down). The inverse transform method [54, 42] is used to determine the size of the jump. Normal variates ($\epsilon \sim N(0,1)$) [42] are used to estimate the standard Brownian motion. The jump and diffusion components are then added to the drift components in the expression for $X(s+\Delta_t)$ to yield the corresponding value for $Y(s+\Delta_t)$.

Algorithm 2.4 (Jump-diffusion processes).

To simulate the jump-diffusion process given by (2.6):

For $s = 0$ *to* t *by* Δ_t

- *Set* $\zeta := -\lambda\left(\frac{p\eta_1}{\eta_1 - 1} + \frac{(1-p)\eta_2}{\eta_2+1} - 1\right)$.
- *Set* $\vartheta := 0$.
- *Generate* $U_1, U_2, U_3 \sim U(0,1)$.
- *If* $(U_1 \leq \lambda\Delta_t)$
  - *If* $(U_2 \leq p), \vartheta = -\frac{\log(U_3)}{\eta_1}$.
  - *Otherwise,* $\vartheta = \frac{\log(U_3)}{\eta_2}$.
- *Generate* $\epsilon \sim N(0,1)$.
- *Set* $X(s+\Delta_t) := (\nu + \zeta - \sigma^2/2)\Delta_t + \vartheta + \sigma\epsilon\sqrt{\Delta_t}$.

- *Set* $Y(s+\Delta_t) := Y(s)\exp(X(s+\Delta_t))$.

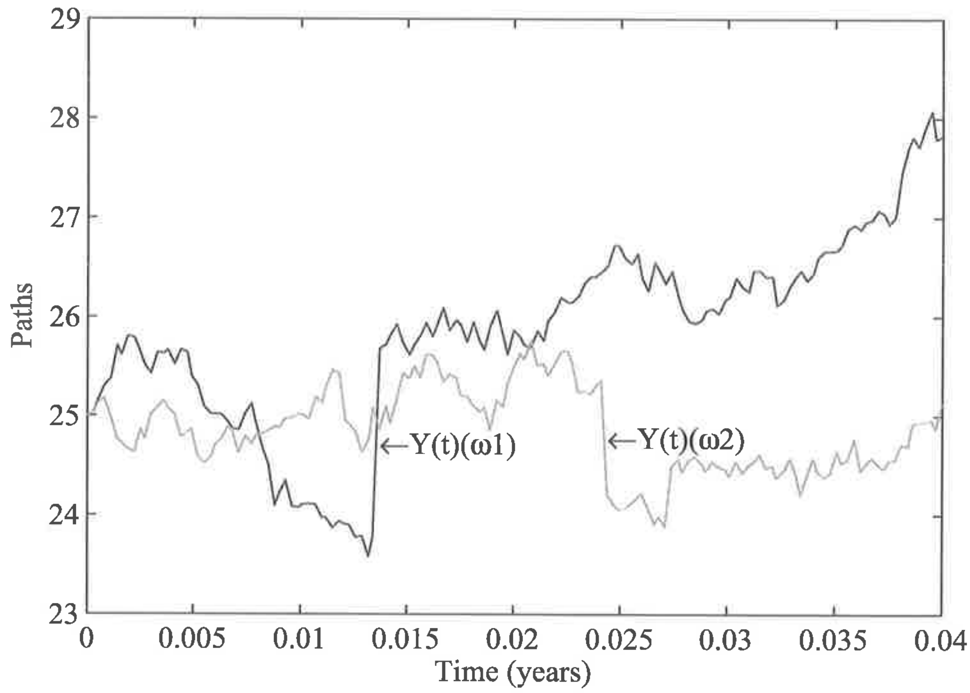


Figure 2.4: Sample paths for a jump-diffusion process

In this thesis, we use a much simpler process which only has jumps in one direction (i.e. downward jumps),

$$dY(t) = \nu Y(t)dt + \sigma Y(t)dB(t) - \phi Y(t)dN(t), \tag{2.7}$$

where ν is the drift, σ is the volatility, ϕ is the jump magnitude, $B(t)$ is a standard Brownian motion, and $N(t)$ is a Poisson process with arrival rate λ. This process has downward jumps provided $\phi > 0$. We require that $1 - \phi > 0$, otherwise the process goes to zero in finite time almost surely. This process is simulated using Algorithm 2.4 and setting $p = 0$, $\eta_1 = 0$ and $\eta_2 = \frac{1-\phi}{\phi}$.

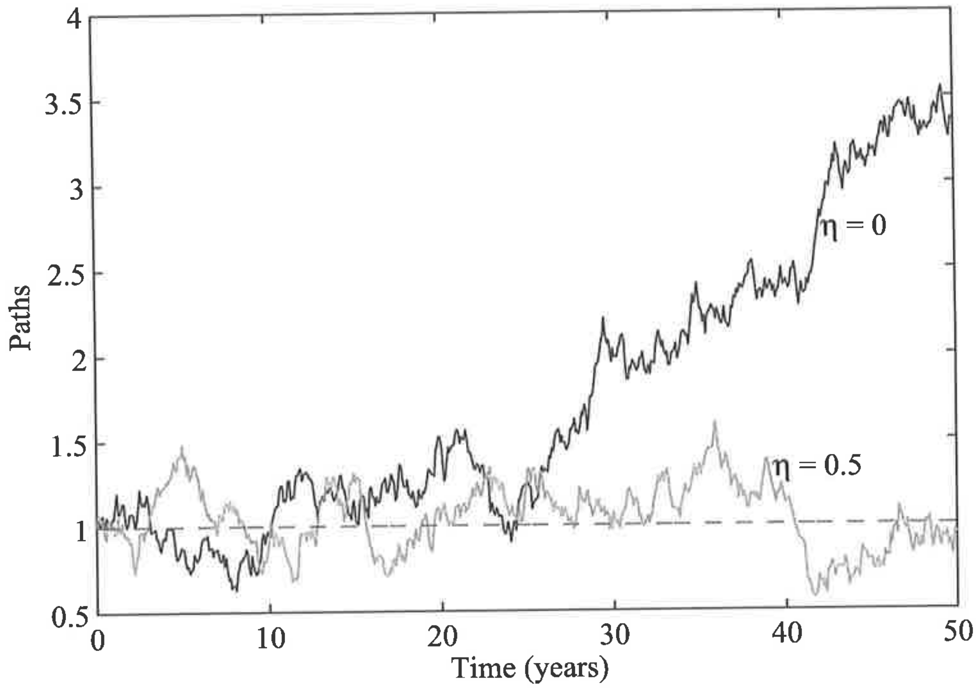


Figure 2.5: Sample paths for an Ornstein-Uhlenbeck process

2.2.3 Mean-Reversion Processes

Mean-reversion processes are used to model raw commodity prices (e.g. copper and oil [22]). The Ornstein-Uhlenbeck process is the simplest mean reverting process,

$$dY(t) = \eta(\overline{Y} - Y(t))dt + \sigma dB(t), \tag{2.8}$$

where η is the speed of reversion, $\overline{Y}$ is the long-run equilibrium level, σ is the volatility and $B(t)$ is a standard Brownian motion. Such processes are called a mean-reversion processes because they keep returning to the equilibrium level $\overline{Y}$: whcnever Y deviates from the equilibrium level, the dt term draws the process back towards the equilibrium level (see Figure 2.5). In this thesis, we use the mean-reversion process known as the logistic model

$$dY(t) = \eta(\overline{Y} - Y(t))Y(t)dt + \sigma Y(t)dB(t). \tag{2.9}$$

Figure 2.6 shows some sample paths for a logistic process. The logistic processes in this thesis were simulated using the Euler method [37].

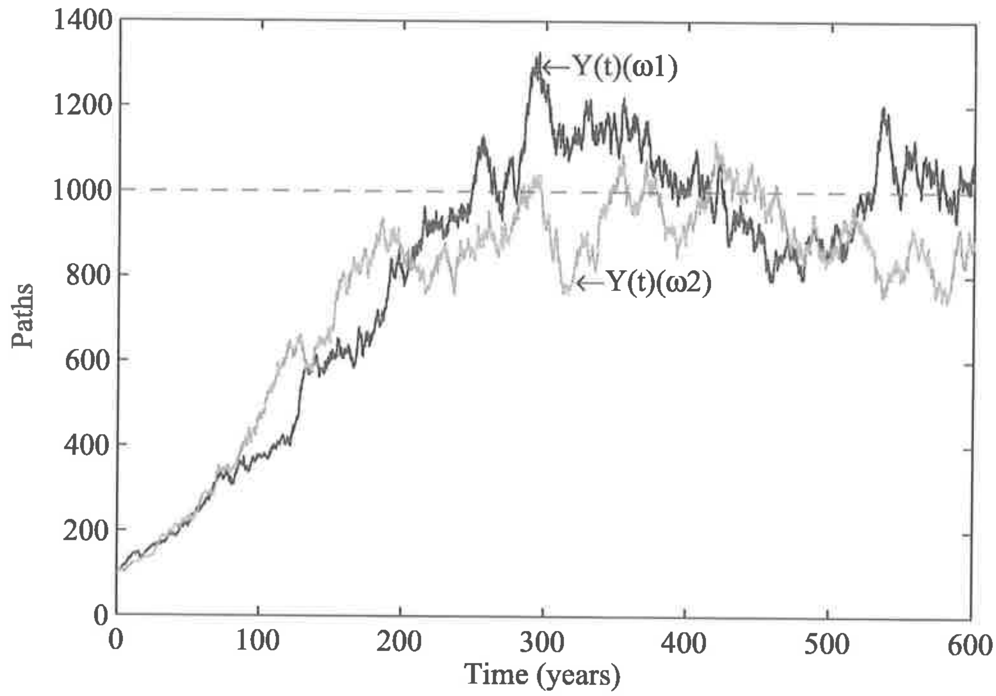


Figure 2.6: Sample paths for a logistic process

Algorithm 2.5 (Logistic processes).

To simulate the logistic process given by (2.9):

For $s = 0$ *to* t *by* Δ_t

- *Generate* $\epsilon \sim N(0,1)$.
- *Set* $Y(s+\Delta_t) := Y(s) + \eta(\overline{Y} - Y(s))Y(s)\Delta_t + \sigma Y(s)\epsilon\sqrt{\Delta_t}$.

2.3 Option Pricing Models

As indicated in Chapter 1 perpetual models will be used to study ICT investments, and the binomial model will be used to determine whether perpetual models are suitable for typical investment durations. Theorem 2.6 provides a solution for the classic perpetual American call model. Algorithm 2.7 provides the pseudo code for a binomial model with N steps. The binomial model will converge to the option value as N increases. Figure 2.7 shows the binomial models for three European call options converging to their respective Black-Scholes values.

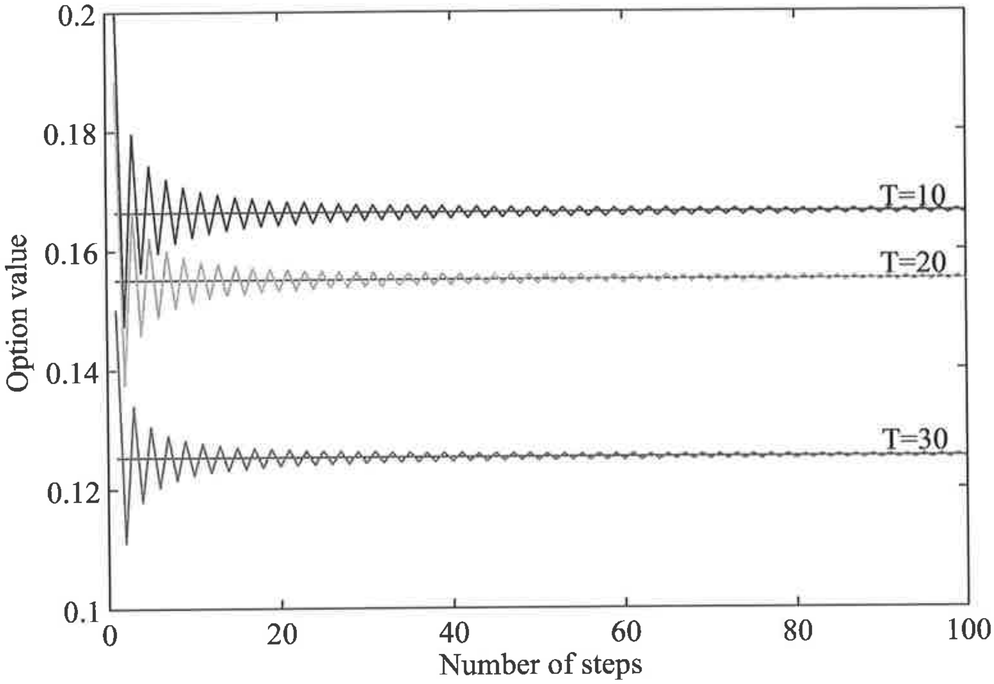


Figure 2.7: The convergence of binomial models

Theorem 2.6 (Perpetual American Call Option [35]).

If the stock price follows a geometric Brownian motion

$$dS(t) = (r - \delta)S(t)dt + \sigma S(t)dB(t),$$

where r is the risk-free rate, δ is the dividend rate, σ is the volality and $B(t)$ is a standard Brownian motion. Then the value of the perpetual American call option with strike price K is

$$F(S) = \begin{cases} (S^* - K)\left(\frac{S}{S^*}\right)^\lambda, & 0 < S < S^*; \\ (S - K), & S \geq S^*, \end{cases}$$

where

$$\begin{aligned} \lambda &= \frac{1}{\sigma}[-v + \sqrt{v^2 + 2r}], \\ v &:= \frac{r - \delta}{\sigma} - \frac{\sigma}{2}, \\ S^* &= \frac{\lambda}{\lambda - 1}K. \end{aligned}$$

Proof. The proof of this theorem is given in [34]. □

Algorithm 2.7 (The Binomial Model).
If the stock price follows a geometric Brownian motion

$$dS(t) = (r - \delta)S(t)dt + \sigma S(t)dB(t),$$

where r is the risk-free rate, δ is the dividend rate, σ is the volality and $B(t)$ is a standard Brownian motion. Then the N-step binomial model for the call option with strike price K operates as follows:

1. *Split the interval $(0, T)$ into N equally spaced time points*

$$\left\{0, \frac{T}{N}, \frac{2T}{N}, \ldots, \frac{(N-1)T}{N}, T\right\}.$$

2. *Set*

$$
\begin{aligned}
u &= \exp\left(\sigma\sqrt{\frac{T}{N}}\right), \\
d &= \exp\left(-\sigma\sqrt{\frac{T}{N}}\right), \\
A &= \exp\left((r-\delta)\frac{T}{N}\right), \\
R &= \exp\left(r\frac{T}{N}\right), \\
p &= \frac{A-d}{u-d}.
\end{aligned}
$$

3. *Construct a tree of prices as follows:*

   (a) *Set* $S(0,0) = S(0)$.

   (b) *For each time* n *in* $\{0,1,\ldots,N\text{-}1\}$:

   For each value $S(n,j)$:

   Assumes that the stock price $S(n,j)$ *will either move up or down at the end of the interval with real-world probabilities* q *and* $1-q$ *respectively:*

   $uS(j,n)$ *with probability* q,

   $S(j,n)$

   $dS(j,n)$ *with probability* $1-q$.

   Since $ud = 1$, *the tree recombines and we have* $(n+1)$ $S(j,n)$ *values at time* n.

4. *Use backward induction to calculate* $V(0,0)$:

   (a) *At the end point* T, *calculate the payoff for each price* $S(j,N)$:

$$V(j,N) = (S(j,N) - K)^{+}. \tag{2.10}$$

   (b) *For each time* $n = \{N-1, N-2, \ldots, 1\}$:

   For each cell (j,n):

We use the risk neutral pricing formula to determine the expected value of future cash flows,

$$E(j,n) = \frac{pV(j+1,n+1) + (1-p)V(j,n+1)}{R}.$$

For European options, the option value is the expected value

$$V(j,n) = E(j,n).$$

For American options, the options can be exercised at any point so we set the option value equal to the maximum of the expected value and payoff.

$$V(j,n) = \max(E(j,n), (S(j,n) - K)^+). \tag{2.11}$$

2.4 Conclusion

This chapter presented some key concepts of stochastic calculus, a description of three stochastic processes that will be used in this thesis, and the technical details for the perpetual American call model and the binomial model. In the next chapter we shall explore two simple models for building new infrastructure.

Chapter 3

Building New Infrastructure

This chapter examines two simple models for building new infrastructure: a geometric Brownian motion (GBM) model and a jump-diffusion process (JDP) model. These models were previously solved using a Partial Differential Equation (PDE) approach [22, 41]. In this chapter, we provide alternative derivations involving martingale methods. We also discuss investment timing issues that were not addressed in the previous literature.

3.1 Introduction

Investment models [43, 22, 41] for building new infrastructure have the same general formulation. An investor, faced with an irreversible investment decision, must decide the optimal time to invest. The firm can pay a sunk cost $I(t)$ to invest in a project whose value at time t of future revenues is $V(t)$. The investment opportunity's final value is the expected profit that the investor receives for investing at time τ, $(V(\tau) - I(\tau))$, discounted by $e^{-r\tau}$ and maximized over the investment (stopping) times τ, i.e.

$$F(V) = \max_{\tau} E_0[(V(\tau) - I(\tau))^+ e^{-r\tau} 1(\tau < \infty)], \tag{3.1}$$

where $E_0[\cdot]$ is the risk-neutral expected value given that $V_0 = V$, $(\cdot)^+$ is the positive part, and $1(\cdot)$ is the indicator function. The optimal strategy is to invest when the value process $V(t)$ first reaches some threshold V^* (which shall be henceforth referred to as the optimal trigger).

As mentioned in Section 1.2, Lassila [41] provides a rationale for using a negative jump-diffusion process to model the value process $V(t)$. However, we shall consider a GBM value process first because geometric Brownian motions are more tractable. As in [22, 41], we assume that the investment cost is fixed $I(t) = I$. Both models were originally solved using a Partial Differential Equation (PDE) approach (see Appendices B.2 and B.3). In Sections 3.2 and 3.3 we solve these models using martingale methods. Regardless of the approach used we need to employ two boundary conditions to obtain an expression for $F(V)$. If the initial value V is zero, $V(t)$ can never move beyond zero and the investment value is also zero. This leads to the initial condition

$$F(0) = 0. \tag{3.2}$$

If the initial value V is greater than the investment trigger V^*, the investment will be made immediately and the investment value is

$$F(V) = V - I,$$

and the smooth pasting condition [21] is

$$F'(V^*) = \frac{d}{dV^*}(V^* - I) = 1. \tag{3.3}$$

The models in [22, 41] are perpetual models and therefore assume that the investment duration (which is called the expiry date in option pricing theory) is very large. It is therefore necessary to verify that the perpetual model is a good approximation for typical investment durations. In Section 3.4, we use a binomial model to see how quickly the finite model converges to the perpetual model. Once we have established that the models are appropriate, we then use stopping times to measure the time when the investment is made (see Section 3.5).

We shall use two examples from [22, 41] as base cases for numerical analysis (see Table 3.1). Example 3.1 is a GBM example from [22] and Example 3.2 is a

Example	ν	σ	r	λ	ϕ	V	I
3.1	0	0.2	0.04	0	0	1	1
3.2	-0.299	0.328	0.1	0.5	0.2	3.5	3.5771
3.2a	-0.299	0.328	0.1	0	0	3.5	3.5771
3.2b	-0.399	0.328	0.1	0	0	3.5	3.5771

Table 3.1: GBM and JDP Value Process Examples

JDP example from [41]. Example 3.2a and Example 3.2b are GBMs with the same parameters as Example 3.2 except that the drift is reduced by $\lambda\phi$ in Example 3.2b.

3.2 A Geometric Brownian Motion Model

In this section we consider a GBM value process and fixed investment costs. McDonald and Siegal [43] give a justification for using geometric Brownian motions. Using a similar approach to Lassila [41] we can show that the value processs will be a GBM when the demand and supply processes are both GBMs. Karatzas and Shreve [35] used a martingale approach to solve an American call option model. We shall use a similar approach to solve this investment problem and then provide some numerical examples.

Suppose that supply, $S(t)$, and demand, $D(t)$, both follow geometric Brownian motions:

$$\begin{aligned} dD(t) &= \nu_D D(t)dt + \sigma_D D(t)dB_D(t), \\ dS(t) &= \nu_S S(t)dt + \sigma_S S(t)dB_S(t), \\ \rho dt &= dB_D(t)dB_S(t), \end{aligned}$$

where ν_D and ν_S are drift terms, σ_D and σ_S are volatility terms, $B_D(t)$ and $B_S(t)$ are standard Brownian motions, and ρ is the correlation between $B_D(t)$ and $B_S(t)$. By the law of supply and demand, the spot price is

$$P(t) = \kappa \frac{D(t)}{S(t)}, \tag{3.4}$$

where κ is a scaling factor. Applying Itô's lemma,

$$\begin{aligned} dP &= \frac{\partial f}{\partial D} + \frac{\partial f}{\partial S} + \frac{1}{2}\left[\frac{\partial^2 f}{\partial D^2}(dD)^2 + 2\frac{\partial^2 f}{\partial D \partial S} dD dS + \frac{\partial^2 f}{\partial S^2}(dS)^2\right] \\ &= \frac{k}{S} dD + \frac{-kD}{S^2} dS + \frac{1}{2}\left[2\frac{-k}{S^2} dD dS + \frac{2kD}{s^3}(dS)^2\right] \\ &= \frac{k}{S} dD - \frac{kD}{S^2} dS - \frac{k}{S^2} dD dS + \frac{kD}{S^3}(dS)^2 \\ &= \frac{kD}{S}\left(\nu_D dt + \sigma_D dB_D(t)\right) - \frac{kD}{S}\left(\nu_S dt + \sigma_S dB_S(t)\right) \\ &\quad - \frac{kD}{S}\left(\rho \sigma_D \sigma_S dt\right) + \frac{kD}{S}\left(\sigma_S^2 dt\right) \\ &= \frac{kD}{S}\left[\left(\nu_D - \nu_S - \rho\sigma_D\sigma_S + \sigma_S^2\right) dt + \sigma_D dB_D(t) - \sigma_S dB_S(t)\right]. \end{aligned}$$

This equation can be re-written as

$$dP(t) = \nu_P P(t) dt + \sigma_P P(t) dB_P(t),$$

where

$$\begin{aligned} \nu_P &= \nu_D - \nu_S + \sigma_S^2 - \rho\sigma_D\sigma_S, \\ \sigma_P &= \sqrt{\sigma_D^2 - 2\rho\sigma_D\sigma_S + \sigma_S^2}, \\ B_P(t) &\equiv \frac{\sigma_D B_D(t) - \sigma_S B_S(t)}{\sqrt{\sigma_D^2 - 2p\sigma_D\sigma_S + \sigma_S^2}}, \end{aligned}$$

and by the Lévy theorem [34], B_P is also a standard Brownian motion. The present value of future revenues $V(t)$ is then given by

$$V(t) = E\left[\int_{t+l}^{\infty} P(s) e^{-r(s-t)} ds\right],$$

where $E[\cdot]$ is the risk-neutral expectation and l is the delay between construction and operation. It was shown in [41] that $V(t)$ has the same dynamics as $P(t)$ (see Appendix B.3), thus

$$dV(t) = \nu_P V(t) dt + \sigma_P V(t) dB_P(t).$$

The investment cost is assumed to be constant (i.e. $I(t) = I$), thus

$$F(V) = \max_{\tau} E_0[(V(\tau) - I)^+ e^{-r\tau} I(\tau < \infty)]. \quad (3.5)$$

Theorem 3.1. *If the revenue process $V(t)$ follows a geometric Brownian motion*

$$dV(t) = \nu V(t)dt + \sigma V(t)dB(t), \tag{3.6}$$

and the cost process is a constant $I(t) = I$. Then the investment value is given by

$$F(V) = \begin{cases} (V^* - I)\left(\frac{V}{V^*}\right)^k, & 0 < V < V^*; \\ (V - I), & V \geq V^*, \end{cases}$$

where

$$\begin{aligned} k &= \frac{1}{\sigma}[-\mu + \sqrt{\mu^2 + 2r}], \\ \mu &= \frac{\nu}{\sigma} - \frac{\sigma}{2}, \\ V^* &= \frac{k}{k-1}I. \end{aligned}$$

Proof. It is shown in [35, pages 65–66] that the optimal time to invest τ^* has the form

$$\tau^* = \inf\{t > 0 | V(t) \geq V^*\}.$$

We note that if $V \geq V^*$, then $\tau^* = 0$ and $F(V) = V - I$. At other times the investment value is

$$\begin{aligned} F(V) &= (V^* - I)E_0[e^{-r\tau^*}I(\tau^* < \infty)] \\ &= (V^* - I)E_0[e^{-r\tau^*}], \end{aligned} \tag{3.7}$$

as $e^{-r\infty} = 0$ when $r > 0$. As $\mu := \frac{\nu}{\sigma} - \frac{\sigma}{2}$, $V(t) = V\exp\{\sigma\mu t + \sigma B(t)\}$. We know that

$$\begin{aligned} V^* &= V(\tau^*) \\ &= V\exp\{\sigma\mu\tau^* + \sigma B(\tau^*)\}. \end{aligned}$$

Raising both sides of this equation to the power k and rearranging we get

$$\exp\{-k\sigma\mu\tau^* - \frac{1}{2}k^2\sigma^2\tau^*\} = \left(\frac{V}{V^*}\right)^k \exp\{k\sigma B(\tau^*) - \frac{1}{2}k^2\sigma^2\tau^*\}. \tag{3.8}$$

From Theorem B.2, we know that $\{\exp(k\sigma B(t) - \frac{1}{2}k^2\sigma^2 t) : t \geq 0\}$ is a martingale. By the optional sampling theorem [53, page 69], we have $E[\exp\{k\sigma B(\tau^*) - \frac{1}{2}k^2\sigma^2\tau^*\}] = 1$ and so taking expectations in (3.8) leads to

$$E[\exp\{-k\sigma v\tau^* - \frac{1}{2}k^2\sigma^2\tau^*\}] = \left(\frac{V}{V*}\right)^k.$$

Choosing $k : \frac{1}{2}\sigma^2 k^2 + \sigma\mu k - r = 0$, then $E[\exp\{e^{-r\tau^*}\}] = (\frac{V}{V*})^k$ and

$$F(V) = (V^* - I)\left(\frac{V}{V*}\right)^k.$$

The quadratic $P_1(k) = \frac{1}{2}\sigma^2 k^2 + \sigma\mu k - r$ has a negative root and a positive root because $P_1(0) = -r < 0$. The positive root will be greater than 1 when $P_1(1) = \nu - r < 0$ (see Figure 3.1). By the initial condition $F(0) = 0$, we are only interested in the positive root $k = \frac{1}{\sigma}[-\mu + \sqrt{\mu^2 + 2r}]$. Finally, the smooth pasting condition $F'(V^*) = 1$ [21] implies that $\frac{k(V^*-I)(V^*)^{k-1}}{(V^*)^k} = 1$ and thus

$$V^* = \frac{k}{k-1}I.$$

□

Setting $\nu = r - \delta$ gives an identical solution to Dixit and Pindyck [22, Chapter 5]. If $\delta \leq 0$, the investment value $F(V) = \max_\tau E_0[(V(\tau) - I)^+ e^{-r\tau} I(\tau < \infty)]$ is infinite and there is no optimal trigger, thus we set $F(V) := 0$ (see Figure 3.2). This behaviour is consistent with that in the American call options with no dividends mentioned in Section 1.1. In finite-time models, the option is exercised at the latest possible moment (i.e. the expiry date) and so the American and European call options are equivalent. In perpetual models, it is always better to wait and so the option is never exercised.

3.3 A Jump-Diffusion Process Model

In the previous section we considered a GBM value process and fixed investment costs. However, Lassila [41] provided a rationale for using a positive JDP to model

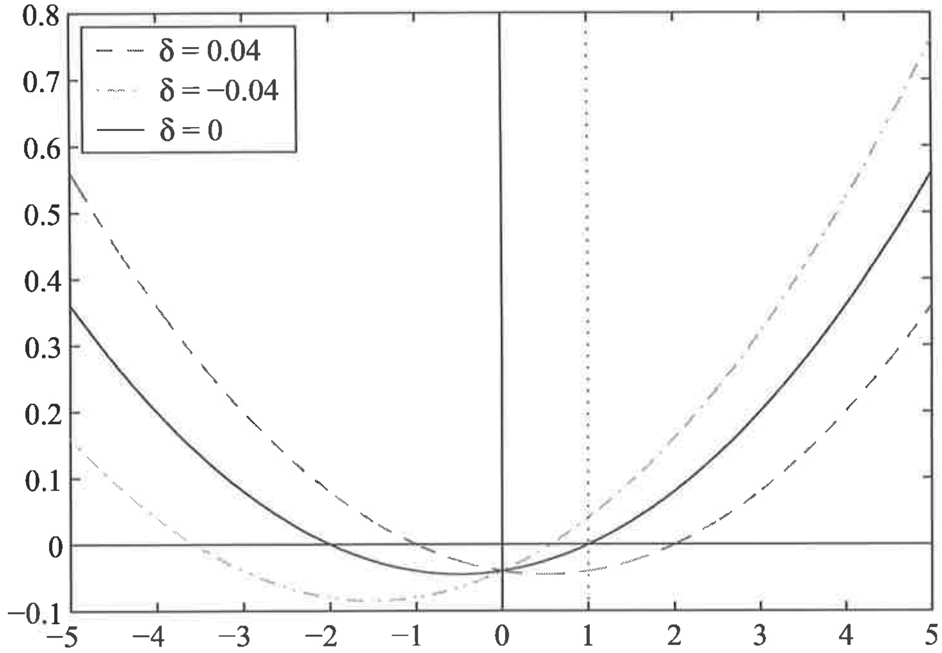


Figure 3.1: The roots of $P_1(k)$

the supply process in bandwidth markets and showed that value process will be a negative JDP when the demand process is a GBM and the supply process is a positive JDP. We shall give these details below (for completeness) before solving the JDP model using martingale methods and providing some numerical analysis.

Suppose that supply, $S(t)$ follows a jump-diffusion process, and demand, $D(t)$, follows a geometric Brownian motions:

$$\begin{aligned} dD(t) &= \nu_D D(t)dt + \sigma_D D(t)dB_D(t), \\ dS(t) &= \nu_S S(t)dt + \sigma_S S(t)dB_S(t) + \phi S(t)dN(t), \\ \rho dt &= dB_D(t)dB_S(t), \end{aligned}$$

where ν_D and ν_S are drift terms, σ_D and σ_S are volatility terms, $B_D(t)$ and $B_S(t)$ are standard Brownian motions, $N(t)$ is a Poisson process with arrival rate λ, and ρ is the correlation between $B_D(t)$ and $B_S(t)$. By the law of supply and demand,

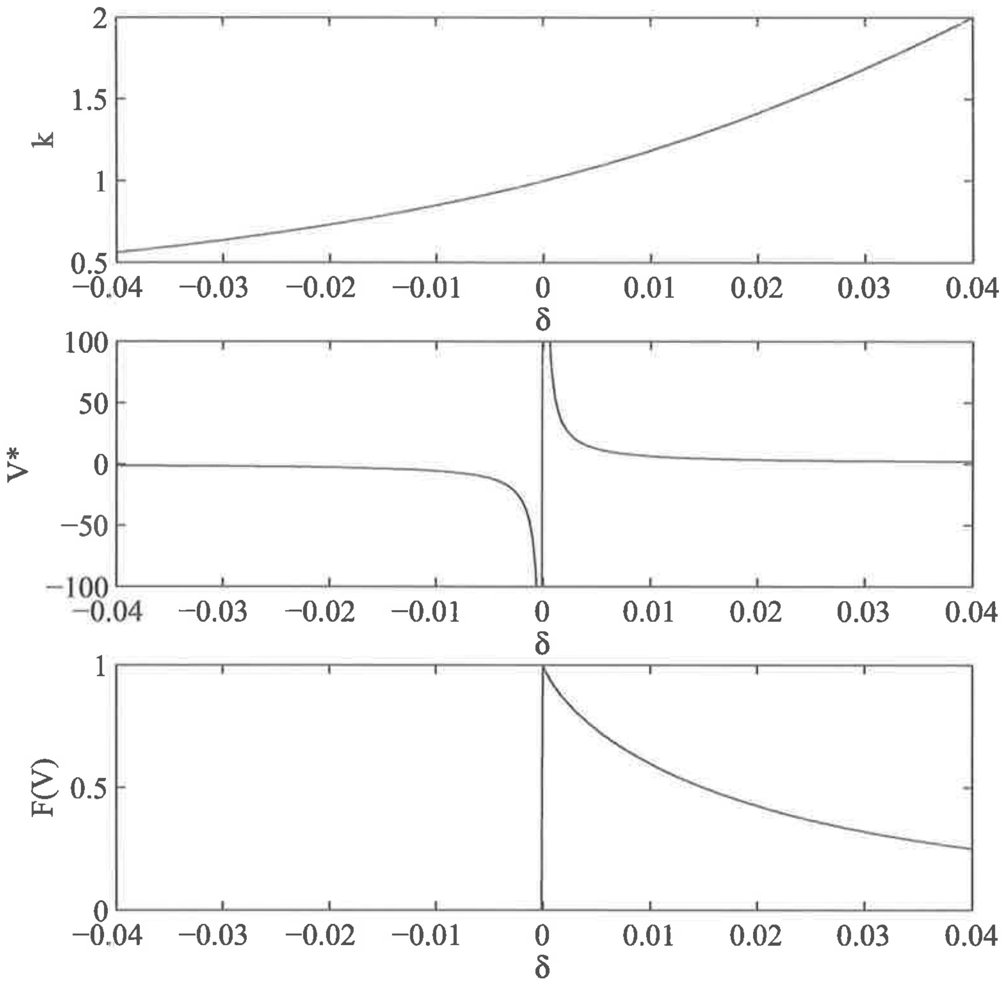


Figure 3.2: A Geometric Brownian Motion model as δ varies

the spot price is

$$P(t) = \kappa \frac{D(t)}{S(t)}, \tag{3.9}$$

where κ is a scaling factor. Applying Itô's lemma (Protter [50, page 71]),

$$\begin{aligned} dP &= \frac{\partial f}{\partial D} + \frac{\partial f}{\partial S} + \frac{1}{2}\left[\frac{\partial^2 f}{\partial D^2}(dD)^2 + 2\frac{\partial^2 f}{\partial D \partial S}dDdS + \frac{\partial^2 f}{\partial S^2}(dS)^2\right] \\ &= \frac{k}{S}dD + \frac{-kD}{S^2}dS + \frac{1}{2}\left[2\frac{-k}{S^2}dDdS + \frac{2kD}{s^3}(dS)^2\right] \\ &= \frac{k}{S}dD - \frac{kD}{S^2}dS - \frac{k}{S^2}dDdS + \frac{kD}{S^3}(dS)^2 \\ &= \frac{kD}{S}\left(\nu_D dt + \sigma_D dB_D(t)\right) - \frac{kD}{S}\left(\nu_S dt + \sigma_S dB_S(t) + \phi dN(t)\right) \\ &\quad -\frac{kD}{S}\left(\rho\sigma_D\sigma_S dt\right) + \frac{kD}{S}\left(\sigma_S^2 dt\right) \\ &= \frac{kD}{S}\left[(\nu_D - \nu_S + \sigma_S^2 - \rho\sigma_D\sigma_S)dt + \sigma_D dB_D(t) - \sigma_S dB_S(t) - \phi dN(t)\right]. \end{aligned}$$

This equation can be re-written as

$$dP(t) = \nu_P P(t)dt + \sigma_P P(t)dB_P(t) - \phi P(t)dN(t), \tag{3.10}$$

where

$$\begin{aligned} \nu_P &= \nu_D - \nu_S + \sigma_S^2 - \rho\sigma_D\sigma_S, \\ \sigma_P^2 &= \sigma_D^2 - 2\rho\sigma_D\sigma_S + \sigma_D^2, \\ B_P(t) &\equiv \frac{\sigma_D B_D(t) - \sigma_S B_S(t)}{\sqrt{\sigma_D^2 - 2p\sigma_D\sigma_S + \sigma_S^2}}. \end{aligned}$$

By the Lévy theorem, B_P is also a standard Brownian motion. We require that $1 - \phi > 0$, otherwise $P(t)$ goes to zero in finite time almost surely.

The present value of future revenues $V(t)$ is then given by

$$V(t) = E\left[\int_{t+l}^{\infty} P(s)e^{-r(s-t)}ds\right],$$

where $E[\cdot]$ is the risk-neutral expectation and l is the delay between construction and operation. It was shown in [41], that $V(t)$ has the same dynamics as $P(t)$, thus

$$dV(t) = \nu_P V(t)dt + \sigma_P V(t)dB_P(t) - \phi V(t)dN(t).$$

The investment cost is assumed to be constant (i.e. $I(t) = I$), thus

$$F(V) = \max_{\tau} E_0[(V(\tau) - I)^+ e^{-r\tau} I(\tau < \infty)]. \tag{3.11}$$

Theorem 3.2. *Suppose that the revenue process is a jump-diffusion process with Poisson arrival rate* λ *and jump magnitude* ϕ

$$dV(t) = \nu V(t)dt + \sigma V(t)dB(t) - \phi V(t)dN(t). \tag{3.12}$$

and the cost process is fixed $I(t) = I$*. Then the investment value is given by*

$$F(V) = \begin{cases} (V^* - I)\left(\frac{V}{V^*}\right)^k, & 0 < V < V^*; \\ (V - I), & V \geq V^*, \end{cases}$$

where k *is the positive root to the equation*

$$\frac{\sigma^2}{2}k(k-1) + \nu k + \lambda(1-\phi)^k - (r+\lambda) = 0, \tag{3.13}$$

and

$$V^* = \frac{k}{k-1}I.$$

Proof. As in Theorem 3.1, we expect the optimal time to invest τ^* to have the form

$$\tau^* = \inf\{t > 0 | V(t) \geq V^*\}.$$

We note that if $V \geq V^*$, then $\tau^* = 0$ and $F(V) = V - I$. At other times the investment value is

$$\begin{aligned} F(V) &= (V^* - I)E_0[e^{-r\tau^*}I(\tau^* < \infty)] \\ &= (V^* - I)E_0[e^{-r\tau^*}], \end{aligned} \tag{3.14}$$

as $e^{-r\infty} = 0$ when $r > 0$. From Theorem B.3, we have $V(t) = V\exp\{X(t)\}$ where

$$X(t) = \left(\nu - \sigma^2/2\right)t + \sigma B(t) + \log(1-\phi)N(t).$$

We know that

$$\begin{aligned} V^* &= V(\tau^*) \\ &= V\exp\{X(\tau^*)\}. \end{aligned} \tag{3.15}$$

Defining

$$g(k) = \frac{\sigma^2}{2}k(k-1) + \nu k + \lambda\left((1-\phi)^k - 1\right),$$

raising both sides of (3.15) to the power k, from each side and rearranging we get

$$\exp\{-g(k)\tau^*\} = \left(\frac{V}{V^*}\right)^k \exp\{kX(\tau^*) - g(k)\tau^*\}, \tag{3.16}$$

From Theorem B.4, we know that $\{\exp(kX(t) - g(k)t) : t \geq 0\}$ is a martingale. By the optional sampling theorem [53, page 69] we have $E[\exp\{kX(\tau^*) - g(k)\tau^*\}] = 1$ and so taking expectations in (3.16) leads to

$$E[\exp\{-g(k)\tau^*\}] = \left(\frac{V}{V^*}\right)^k.$$

Choosing $k : g(k) = r$, then $E[\exp\{e^{-r\tau^*}\}] = (\frac{V}{V^*})^k$ and

$$F(V) = (V^* - I)\left(\frac{V}{V^*}\right)^k.$$

The function $P_2(k) = \frac{\sigma^2}{2}k(k-1) + \nu k + \lambda(1-\phi)^k - (r+\lambda)$ has a negative root and a positive root because $P_2(0) = -r < 0$. The positive root will be greater than 1 when $P_2(1) = \nu - r - \lambda\phi < 0$ (see Figure 3.3). By the initial condition $F(0) = 0$, we are only interested in the positive root of $P_2(k)$. Finally, the smooth pasting condition $F'(V^*) = 1$ [21] implies that $\frac{k(V^*-I)(V^*)^{k-1}}{(V^*)^k} = 1$ and thus

$$V^* = \frac{k}{k-1}I.$$

□

This solution is equivalent to the model derived in Lassila [41, Chapter 5]. If $\delta = r - \nu \leq -\lambda\phi$, the investment value $F(V) = \max_\tau E_0[(V(\tau)-I)^+ e^{-r\tau} I(\tau < \infty)]$ is infinite and there is no optimal trigger, thus we set $F(V) := 0$ (see Figure 3.4). The geometric Brownian motion model, presented in the previous section is a special case of the jump-diffusion model. We can obtain the GBM model by setting either $\phi = 0$ or $\lambda = 0$. Figure 3.5 shows the characteristic equations for Examples 3.2, 3.2a and 3.2b. Since the Poisson process effectively reduces the drift by $\lambda\phi$, Example 3.2b is a

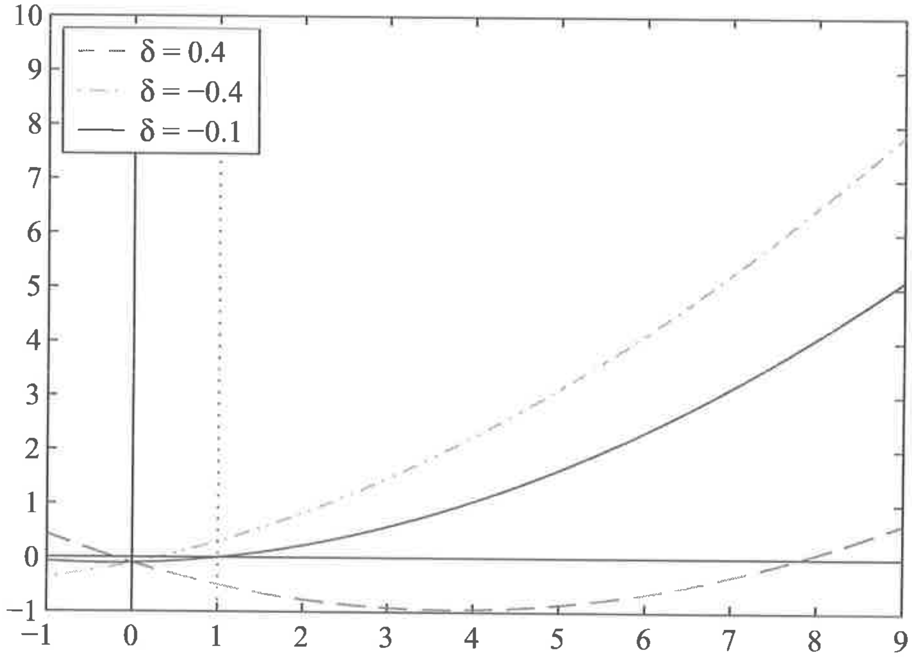


Figure 3.3: The positive roots of $P_2(k)$

better approximation for Example 3.2 than Example 3.2a. The graph also suggests that the characteristic equation for Example 3.2 is bounded by the characteristic equations for Examples 3.2a and 3.2b when $k > 1$.

Proposition 3.3. *The characteristic equation for the JDP model is bounded by the characteristic equations for two GBM models (with the same parameters as the JDP model except that the drift is reduced by $\lambda\phi$ in the second GBM model) for $k \in (1, \infty)$ and $0 < \phi < 1$, viz. $\widetilde{P}_1(k) \leq P_2(k) \leq P_1(k)$ where $\widetilde{P}_1(k)$, $P_2(k)$ and $P_1(k)$ are the characteristic equations for the second GBM, JDP, and first GBM models respectively.*

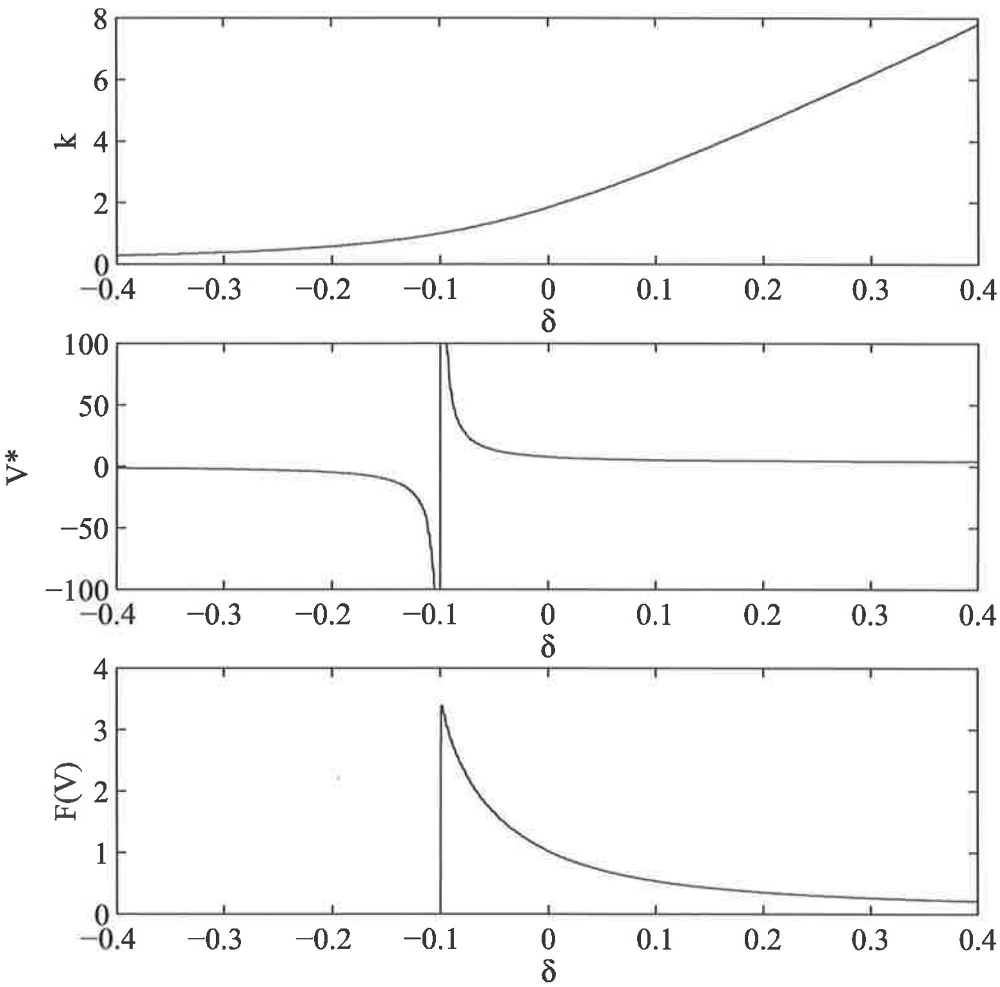


Figure 3.4: A Jump-Diffusion model as δ varies

Proof. Let $f(k) = P_2(k) - \widetilde{P}_1(k) = \lambda\phi k + \lambda(1-\phi)^k - \lambda$

$$\begin{aligned} f(1) &= 0 \\ f'(k) &= \lambda k - \lambda k(1-\phi)^{k-1} \\ &= \lambda k(1-(1-\phi)^{k-1}) \geq 0. \end{aligned}$$

Thus we have $f(k) \geq 0$ and $P_2(k) \geq \widetilde{P}_1(k)$.

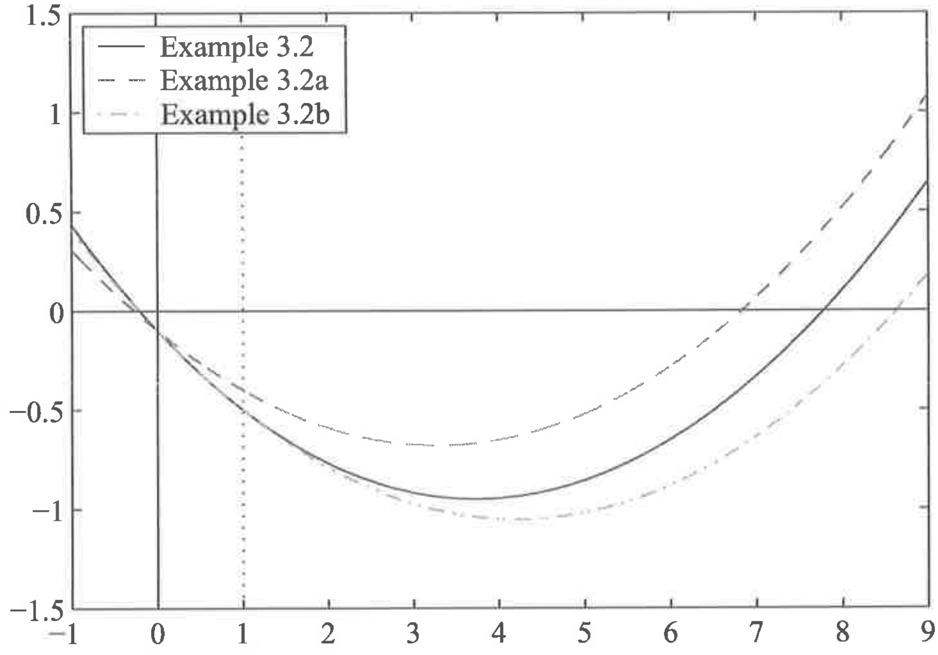


Figure 3.5: The characteristic equations for Example 3.2, 3.2a and 3.2b

Let $g(k) = P_2(k) - P_1(k) = \lambda(1-\phi)^k - \lambda$

$$\begin{aligned} g(0) &= 0 \\ g'(k) &= -\lambda k(1-\phi)^{k-1} \leq 0. \end{aligned}$$

Thus we have $g(k) \leq 0$ and $P_2(k) \leq P_1(k)$. □

Corollary 3.4. *The positive root for the jump model is bounded by the positive roots for two GBM models for $k \in (1, \infty)$ and $0 < \phi < 1$, viz. $\Lambda_1(\nu, r) \leq \Lambda_2(\nu, r) \leq \Lambda_1(\nu - \lambda\phi, r)$ where $\Lambda_1(\nu, r)$, $\Lambda_2(\nu, r)$ and $\Lambda_1(\nu - \lambda\phi, r)$ are the positive roots for the first GBM, JDP, and second GBM models respectively.*

Proof. Let $k_1 := \Lambda_2(\nu, r)$. By Proposition 3.3,

$$\widetilde{P}_1(k_1) \leq P_2(k_1) = 0$$

and so

$$\Lambda_1(\nu - \lambda\phi, r) \geq k_1 = \Lambda_2(\nu, r).$$

Let $k_2 := \Lambda_1(\nu, r)$. By Proposition 3.3,

$$P_2(k_2) \leq P_1(k_2) = 0$$

and so

$$\Lambda_2(\nu, r) > k_2 = \Lambda_1(\nu, r).$$

□

3.4 Convergence of Finite Models

In the previous sections, we developed two simple investment models. But these models assume large investment durations (expiry dates). It is therefore of interest to determine whether the perpetual model is a good approximation for typical investment durations. The binomial model, introduced in Section 2.3, was used to observe the rate at which the finite model converged to the perpetual model. The binomial model for Example 3.1 was initially tested with investment duration $T = 150$ and step size $N = 150$. This result turned out to be inaccurate because the step size was too small relative to the expiry date. In subsequent tests we were able to achieve far greater accuracy for much smaller expiry dates (e.g. $T = 60$) when we used $N = 5T$. Figure 3.6 shows the rate of convergence for Example 3.1. This graph suggests that perpetual model is not a good approximation for Example 3.1 when the investment duration less than 60 years.

Figure 3.7 shows the rate of convergence for Examples 3.2a and 3.2b. This graph suggests that these examples converge much more quickly than Example 3.1; the binomial model converges to the perpetual model in ten years. However, the upper limit for the binomial tree value for $N = 5T$ is significantly smaller than the perpetual value. Choosing $N = 50T$ gives better accuracy and we could continue to improve the accuracy by choosing finer interval lengths. Since the investment value

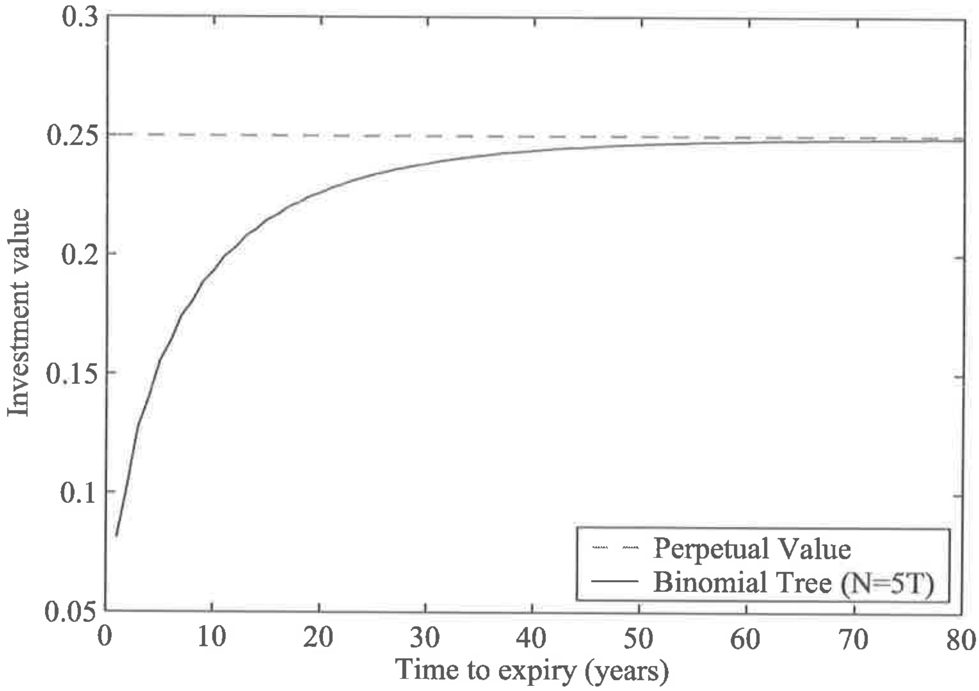


Figure 3.6: The convergence of finite models for Example 3.1

for Example 3.2 is between that of these two examples, we expect that the perpetual model is a good approximation for Example 3.2 when the investment duration is greater than ten years.

3.5 Stopping Times

In the previous section, we determined whether the perpetual model is a good approximation for typical investment durations. In this section we use stopping times to determine whether and when the investment will be made. Geometric Brownian motions are considered separately because they have exact solutions for the stopping times. In general, however, the stopping times must be estimated using simulation techniques.

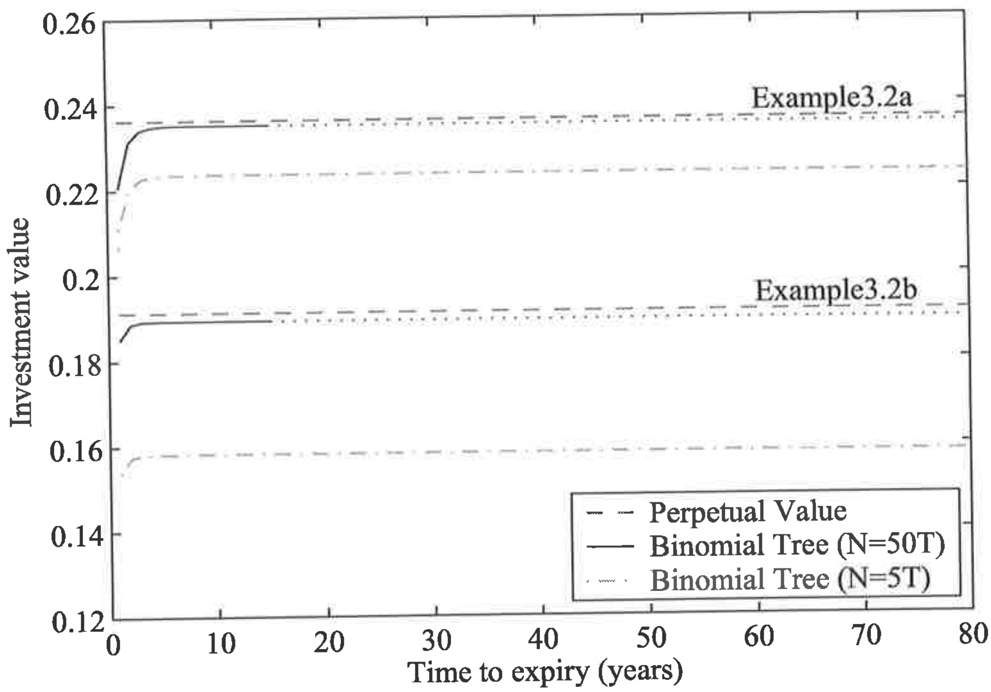


Figure 3.7: The convergence of finite models for Example 3.2a and 3.2b

3.5.1 Geometric Brownian Motions

A stopping time

$$\tau^* = \inf\{t > 0 | V(t) \geq V^*\},$$

where $V(t)$ is a geometric Brownian motion, can be re-written in terms of a Brownian motion $X(t) = \mu t + B(t)$ with $\mu = \frac{\nu}{\sigma} - \frac{\sigma}{2}$,

$$\begin{aligned}
\tau^* &= \inf\{t > 0 | V(t) \geq V^*\} \\
&= \inf\{t > 0 | V \exp\{\sigma X(t)\} \geq V^*\} \\
&= \inf\left\{t > 0 | X(t) \geq \frac{1}{\sigma} \log\left(\frac{V^*}{V}\right)\right\} \\
&= \inf\{t > 0 | X(t) \geq m\},
\end{aligned}$$

where

$$m = \frac{1}{\sigma} \log\left(\frac{V^*}{V}\right).$$

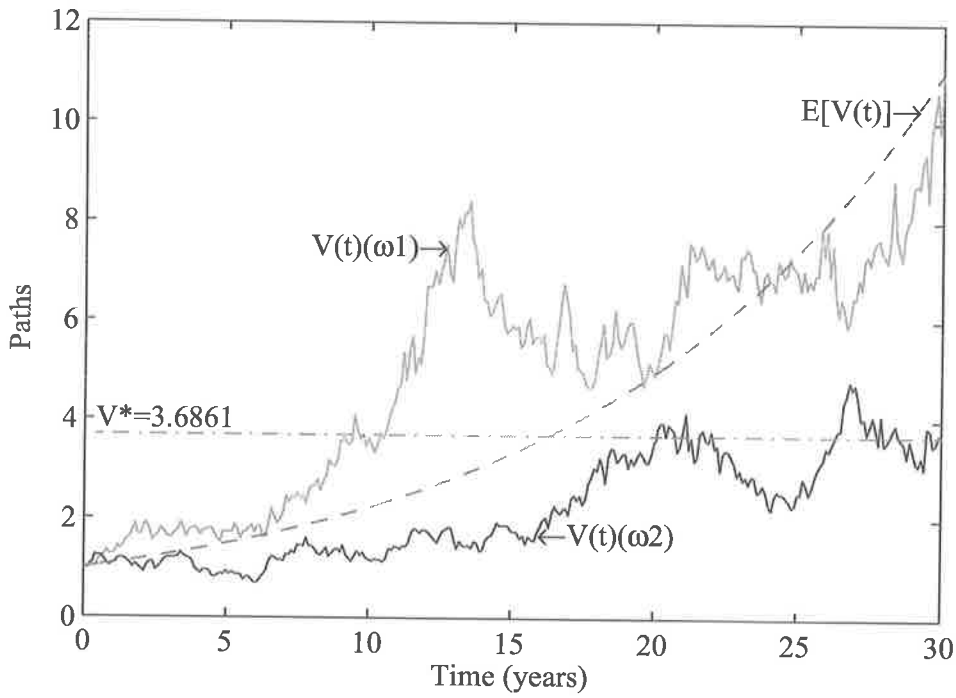


Figure 3.8: Sample paths for $V(t) = \exp\{0.06t + 0.2B(t)\}$

Transforming the stopping time for Geometric Brownian motion into a stopping time for a Brownian motion, allows us to use the formulae provided in Section 2.1.3. From (2.4), we know that the stopping time is almost surely finite when $\mu = \nu - \frac{1}{2}\sigma^2$ is non-negative. Furthermore, (2.3) gives us the probability that the investment occurs before some time t, $P(\tau^* \leq t)$. Thus we require $\nu \geq \frac{\sigma^2}{2}$ for guaranteed investment. Figures 3.8 and 3.9 show typical sample paths for $\mu > 0$ and $\mu < 0$, respectively.

We would also like to calculate the real-world expected stopping times, but these depend on the market price of risk which is often difficult to determine. We shall therefore calculate the risk-neutral expected stopping times. From earlier, we know that the stopping time will always be zero if the initial value of $V(t)$ is greater than or equal to the optimal trigger (i.e. $V \geq V^*$). For $V < V^*$, the average stopping time is only finite when $\mu > 0$ [56]. Thus we seek an expression for $E[\tau^*]$ when

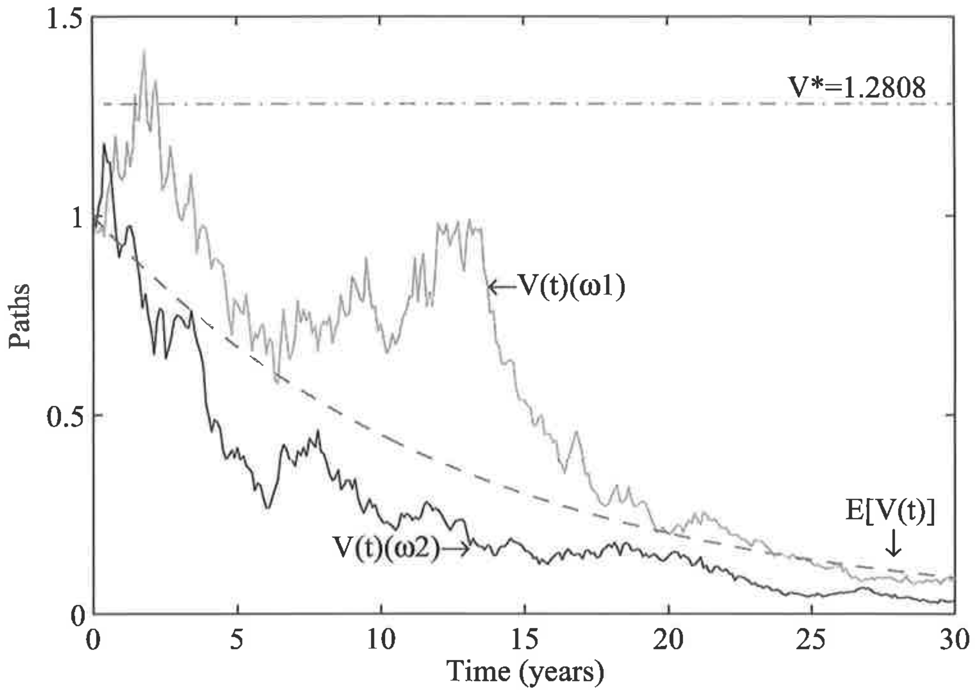


Figure 3.9: Sample paths for $V(t) = \exp\{-0.1t + 0.2B(t)\}$

$\mu > 0$. From earlier, choosing λ such that $\frac{1}{2}\sigma^2\lambda^2 + \sigma\mu\lambda - \beta = 0$, we have

$$\begin{aligned} E[e^{-\beta\tau^*}] &= \left(\frac{V}{V*}\right)^{\frac{1}{\sigma}(-\mu+\sqrt{\mu^2+2\beta})} \\ &= e^{-m(-\mu+\sqrt{\mu^2+2\beta})}. \end{aligned}$$

Taking the derivative with respect to β,

$$E[-\tau^* e^{-\beta\tau^*}] = \frac{-m}{\sqrt{\mu^2+2\beta}} e^{-m(-\mu+\sqrt{\mu^2+2\beta})}.$$

Setting $\beta = 0$ gives,

$$E[\tau^*] = \frac{m}{\mu}. \tag{3.17}$$

It is also useful to calculate the expected stopping time given that the stopping time is less than t,

$$E[\tau^*|\tau^* \leq t] = \frac{\int_0^t sg(s)ds}{G(t)},$$

where

$$G(t) = \int_0^t g(s)ds = P(\tau^* \leq t).$$

We can estimate it using integration by parts,

$$E[\tau^*|\tau^* \leq t] = \frac{[tG(s)]_0^t - \int_0^t G(s)ds}{G(t)}. \quad (3.18)$$

Since there is no closed form solution, numerical integration techniques (e.g. Simpson's rule) can be employed.

3.5.2 General Stochastic Processes

In general, there are no exact solutions for the stopping times. However, the probability that the investment will be made before some time t, $P(\tau < t)$, and the conditional expectation, $E(\tau|\tau < t)$, can be estimated by simulating the revenue process up to time t, N times,

$$P(\tau^* \leq t) \approx \frac{\sum_{j=1}^N 1(\tau_j^* \leq t)}{N}. \quad (3.19)$$

$$E[\tau^*|\tau^* \leq t] \approx \frac{\sum_{j=1}^N \tau_j^* 1(\tau_j^* \leq t)}{\frac{1}{N}\sum_{j=1}^N 1(\tau_j^* \leq t)}. \quad (3.20)$$

Furthermore, when t is chosen sufficiently large, we have

$$P(\tau < \infty) \approx P(\tau \leq t),$$

and if τ^* is almost surely finite (i.e. $P(\tau < \infty) = 1$), then

$$E[\tau] \approx E[\tau^*|\tau^* \leq t].$$

3.5.3 Numerical Results

Figures 3.10 and 3.11 show the stopping times for Examples 3.1 and 3.2a. Since Example 3.2a and 3.2b only differ by δ, Figure 3.11 also shows the stopping times for Example 3.2b. Since these examples are GBM models, we can use the exact

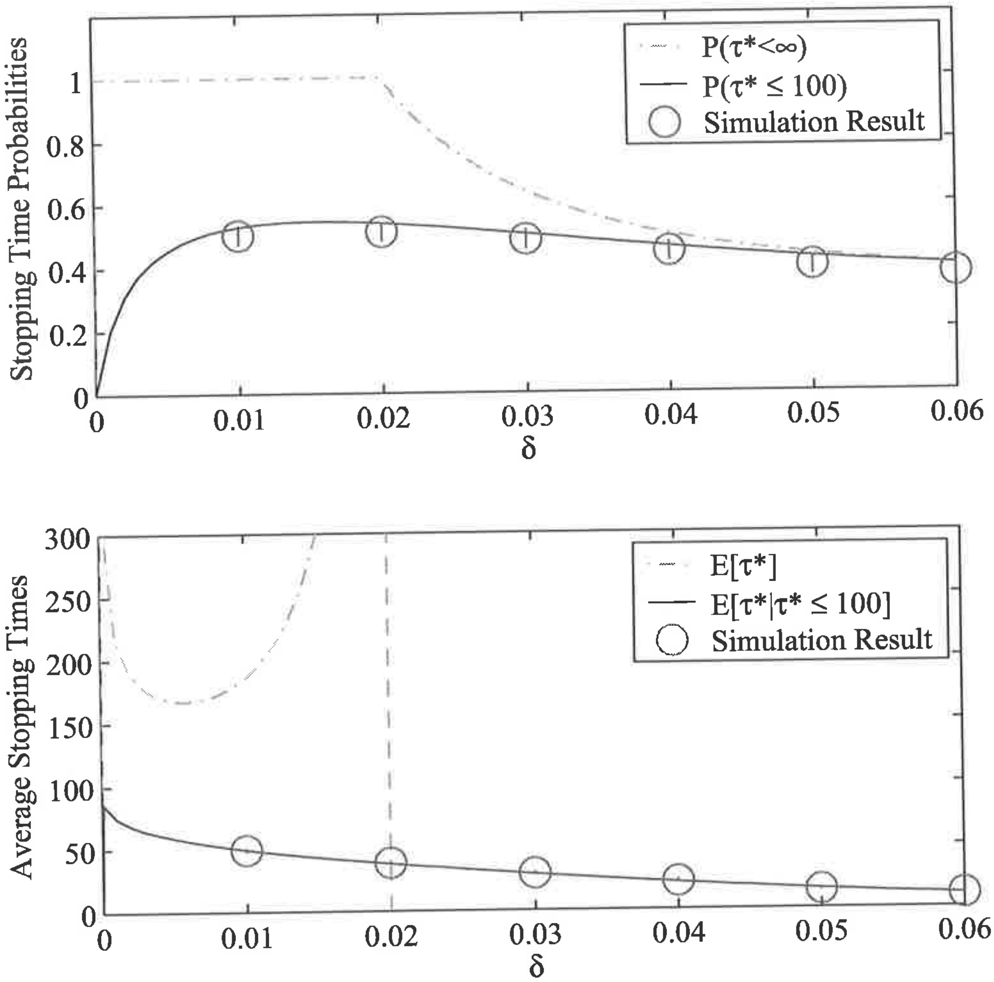


Figure 3.10: Stopping times for Example 3.1 as δ varies

formulae given in Section 3.5.1. The processes were also simulated for $N = 1000$ runs at selected points. The 95% confidence intervals are shown in both graphs. As mentioned in Section 3.2, the investment will never be made if $\delta \leq 0.0$, so there is no stopping time. We note that the investment is guaranteed for $\delta \in (0, 0.02)$ and $(0, 0.0462)$ respectively.

Figure 3.12 shows the stopping times for Example 3.2. Since Example 3.2 is a

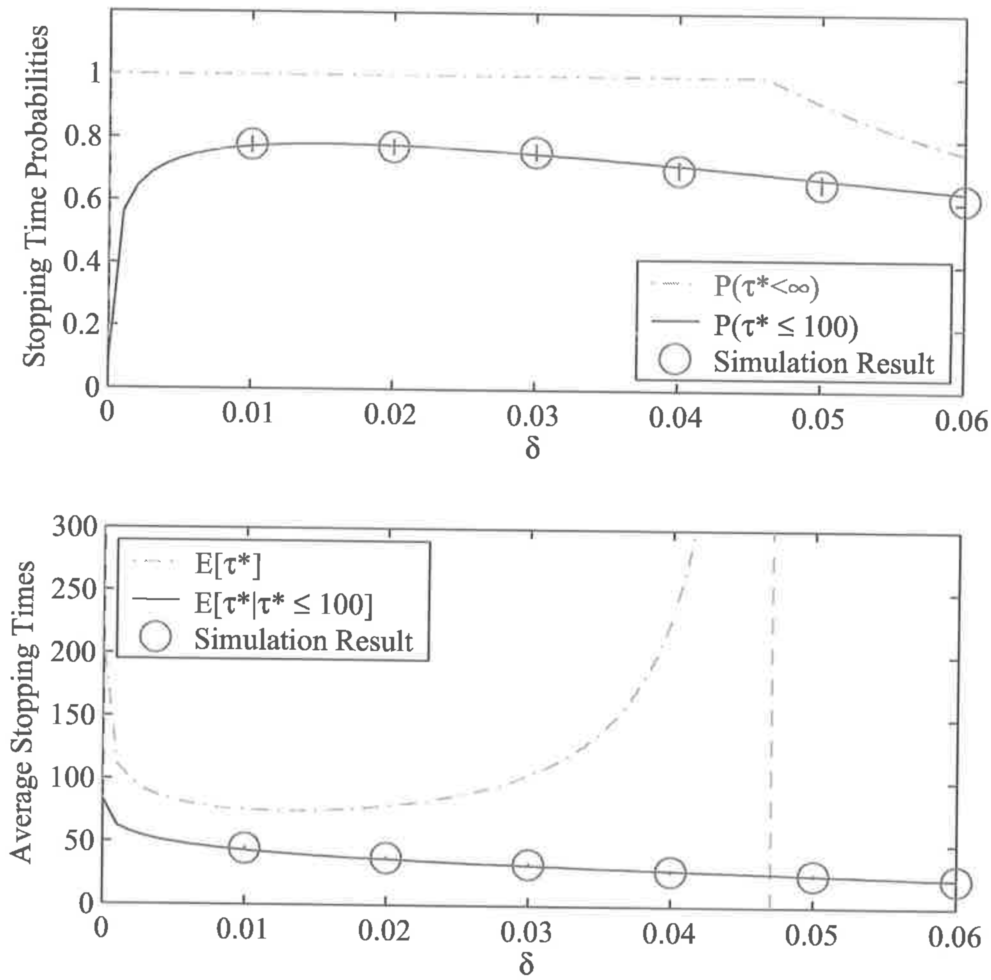


Figure 3.11: Stopping times for Examples 3.2a and 3.2b as δ varies

JDP model we cannot use the exact formulae given in Section 3.5.1. Instead the graphs connect the sample means for simulation results, with $N = 1000$ runs and $t = 100$ or $t = 1000$, at selected values of δ. As mentioned in Section 3.3, the investment is never made when $\delta < -\lambda\phi = -0.1$. The upper graph suggests that the investment is guaranteed for $\delta \in (-0.1, 0)$ but it is difficult to determine the first value for which $P(\tau^* < \infty)$ ceases to be equal to one.

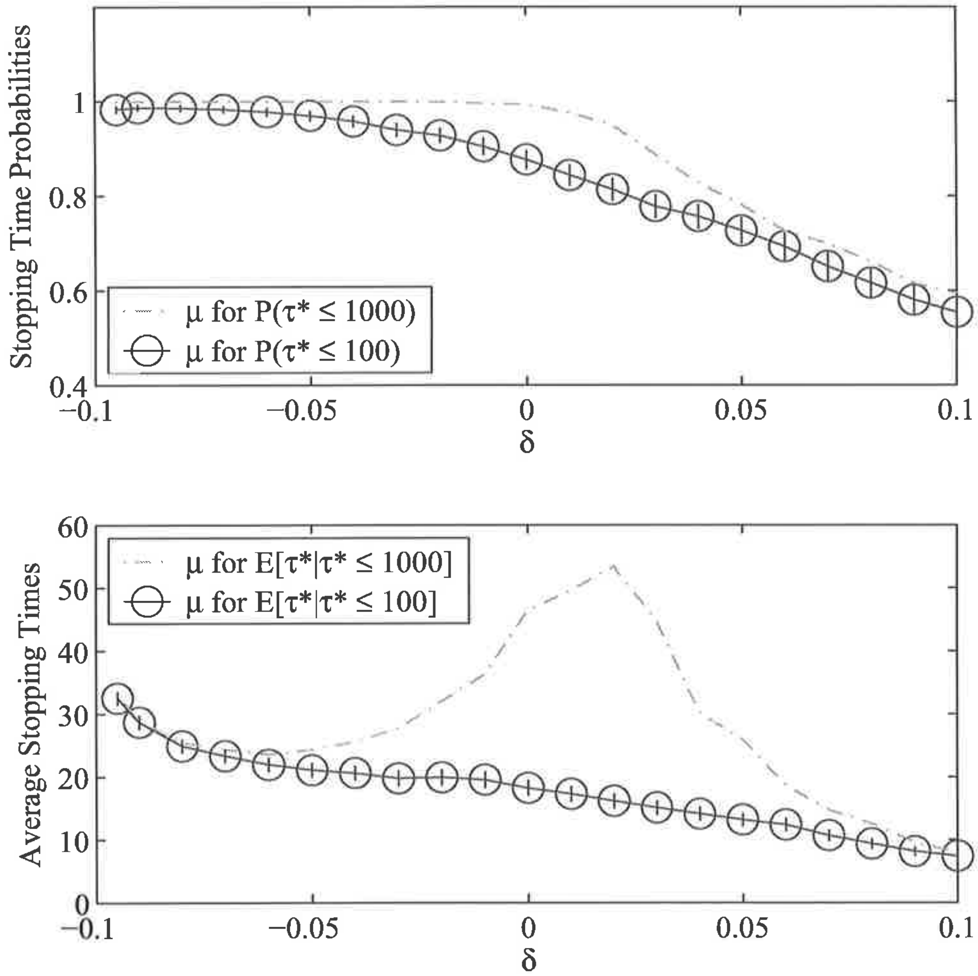


Figure 3.12: Stopping times for Example 3.2 as δ varies

3.6 Conclusion

In this chapter, we presented two simple models for building new infrastructure. These models were previously solved using a PDE approach. We provided an alternative derivation using martingale methods. Since these models are perpetual models the investment duration is assumed to be very large. The binomial model was used to determine whether the perpetual model is a good approximation for typical investment durations. Stopping times were used to determine whether and when the investment will be made. In the next chapter we shall extend these models to support decreasing investment costs and investigate common error scenarios.

Chapter 4

Decreasing Investment Costs

In the previous chapter we presented two simple models for building new infrastructure: a geometric Brownian motion (GBM) model and a jump diffusion process (JDP) model. These models assume that investment costs are fixed. In the Information and Communication Technology (ICT) industry, however, we expect investment costs to decrease exponentially according to Moore's and Gilder's laws. This chapter extends the GBM and JDP models to decreasing investment costs and studies the impact of common error scenarios.

4.1 Introduction

The dominant feature of investments in the ICT industry is that following Moore's law, the cost of investment decreases approximately exponentially over time. In 1965, Moore predicted that the number of transistors per chip would double each year for the next ten years [45]. At the end of the period, he predicted that the the capacity per chip would increase by a factor of two every 18 months [48]. Regardless of the decay parameter chosen, these predictions suggest exponential decay in costs (for a computer of the same power).

Similar relationships to Moores' law also apply in the telecommunications industry. Gilder [27, 26] predicted that bandwidth would triple each year for the next

25 years. Although Coffman and Odlyzko [48] believe that these predictions are exaggerated, their own projections suggest that the transmission capacity of each fiber will increase by a factor of two each year.

This chapter is concerned with quantifying investment decisions in this context. In Section 4.2 we develop a model for valuing investment decisions that allows for declining investment costs. This model assumes that the true decay parameter is known. However, the above literature suggests that this is seldom the case. Section 4.3 develops an error model for the decay parameter (which we shall henceforth call the cost error model) and explores a variety of error scenarios.

Several authors have suggested that flawed analysis of internet growth contributed to the Internet bubble. Coffman and Odlyzko [15] suggest that the expectation of data traffic doubling every few months (rather than the more realistic estimates of doubling every year) led individuals and companies to invest inappropriately. For example, in North America more than half a dozen long-haul carriers laid down enough optical fiber to provide much more capacity than was really needed [15]. In Section 4.4 we provide a brief overview of how traffic errors may be analyzed and explain how this error model differs from the cost error model.

Lemma 4.1 defines a general formula which encompasses the geometric Brownian motion and jump-diffusion process models described in Chapter 3. This formula will be useful in Sections 4.3-4.4. By defining the more sophisticated models in terms of this general formula we obtain solutions for both models.

Lemma 4.1. *If the revenue process $V(t)$ follows a geometric Brownian motion or jump-diffusion process given by (3.6) or (3.12) respectively, and the cost process is a constant $I(t) = I$, then the investment value is given by*

$$F(V) = \begin{cases} (V^* - I)\left(\frac{V}{V^*}\right)^k, & 0 < V < V^* \\ (V - I), & V \geq V^* \end{cases}$$

where

$$\begin{aligned} k &= \Lambda(\nu, r), \\ V^* &= \frac{k}{k-1} I. \end{aligned}$$

If the revenue process follows a geometric Brownian motion, $\Lambda(\nu, r)$ denotes the positive root of

$$P_1(k) = \frac{1}{2}\sigma^2 k^2 + \sigma\mu k - r,$$

i.e.

$$\Lambda(\nu, r) = \frac{-\left[\nu - \frac{\sigma^2}{2}\right] + \sqrt{\left(\nu - \frac{\sigma^2}{2}\right)^2 + 2r\sigma^2}}{\sigma^2}.$$

If the revenue process follows a jump-diffusion process, $\Lambda(\nu, r)$ denotes the positive root of

$$P_2(k) = \frac{\sigma^2}{2} k(k-1) + \nu k + \lambda(1-\phi)^k - (r + \lambda).$$

Proof. Combining Theorems 3.1 and 3.2 gives the generic formula. □

The examples in Table 3.1 will be used as base cases for numerical analysis. Section 4.2 investigates the behaviour of Example 3.1 and 3.2 as the decay parameter increases. In Sections 4.3 and 4.4 we restrict our analysis to the original GBM example (Example 3.1) because the GBM model is more tractable than the JDP model. We expect the two models to behave in a similar fashion, and our results in Section 4.2 suggest that the JDP model does not provide much advantage over the GBM model when the decay parameter is large.

4.2 An Investment Model with Decreasing Costs

The literature suggests that chip capacity and bandwidth capacity are increasing exponentially over time or conversely costs are decreasing. We therefore expect investment costs to decrease over time and adopt a negative exponential function for the cost function

$$I(t) = Ie^{-\alpha t}. \tag{4.1}$$

For decreasing investment costs, the decay parameter α will be positive. In this section, we use (4.1) to develop a general formula for the cost model and investigate the behaviour of various parameters in the GBM and JDP models.

4.2.1 The General Formula

Substituting (4.1) in (3.1) yields

$$F(V) = \max_{\tau} E[(V(\tau) - Ie^{-\alpha t})^{+} e^{-r\tau} | V(0) = V]. \tag{4.2}$$

This expression can be re-written as

$$F(V) = \max_{\tau} E[(Y(\tau) - I)^{+} e^{-(r+\alpha)\tau} | Y(0) = V],$$

where $Y(t) = V(t)e^{\alpha t}$. This is equivalent to increasing ν and r by α, and leads to the following lemma. Note that the formula is the same as that given in Lemma 4.1 except that $k = \Lambda(\nu, r)$ is replaced by $k = \Lambda(\nu + \alpha, r + \alpha)$.

Lemma 4.2. *If the revenue process $V(t)$ follows a geometric Brownian motion or jump-diffusion process given by (3.6) or (3.12) respectively, and the cost process is $I(t) = Ie^{-\alpha t}$, then the investment value is given by*

$$F(V) = \begin{cases} (V^* - I)\left(\frac{V}{V^*}\right)^k, & 0 < V < V^*; \\ (V - I), & V \geq V^*, \end{cases}$$

where

$$\begin{aligned} k &= \Lambda(\nu + \alpha, r + \alpha), \\ V^* &= \frac{k}{k-1} I. \end{aligned}$$

If the revenue process follows a geometric Brownian motion, $\Lambda(\nu + \alpha, r + \alpha)$ denotes the positive root of

$$P_1^{\alpha}(k) = \frac{1}{2}\sigma^2 k(k-1) + (\nu + \alpha)k - (r + \alpha),$$

i.e.

$$\Lambda(\nu+\alpha, r+\alpha) = \frac{-\left[\nu+\alpha-\frac{\sigma^2}{2}\right] + \sqrt{\left(\nu+\alpha-\frac{\sigma^2}{2}\right)^2 + 2(r+\alpha)\sigma^2}}{\sigma^2}.$$

If the revenue process follows a jump-diffusion process, $\Lambda(\nu, r)$ denotes the positive root of

$$P_2^\alpha(k) = \frac{\sigma^2}{2}k(k-1) + (\nu+\alpha)k + \lambda(1-\phi)^k - (r+\alpha+\lambda).$$

Proof. Replacing ν with $\nu+\alpha$ and r with $r+\alpha$ in Corollary 4.1 gives the formula. □

Note that it will be convenient below to write $\Lambda = \Lambda_1, \Lambda_2$ for these two cases. On the surface, it would appear that the general formula also applies to increasing investment costs (i.e. $\alpha < 0$). However, the proofs given in Sections 3.2 and 3.3 assumed $r > 0$, and so we require $r + \alpha > 0$ to get the formula. This means that Lemma 4.2 only applies for small negative values (i.e. $\alpha > -r$).

4.2.2 The Positive Root

Figure 4.1 shows the positive roots for Example 3.1 as the decay parameter, α, varies on $(0, 6)$. Figure 4.2 shows the positive roots for Example 3.2 as α varies on (0,4). Since the positive root was greater than one in the original (fixed-cost) examples, they will remain greater than one in the decreasing cost examples because the relationship between the drift rate ν and the interest rate r (i.e. $\nu < r$) is preserved when they are both increased by α. However, the graphs suggest that k is converging to one as α increases.

Lemma 4.3. *The positive root $k := \Lambda(\nu+\alpha, r+\alpha)$ converges to one as $\alpha \to \infty$.*

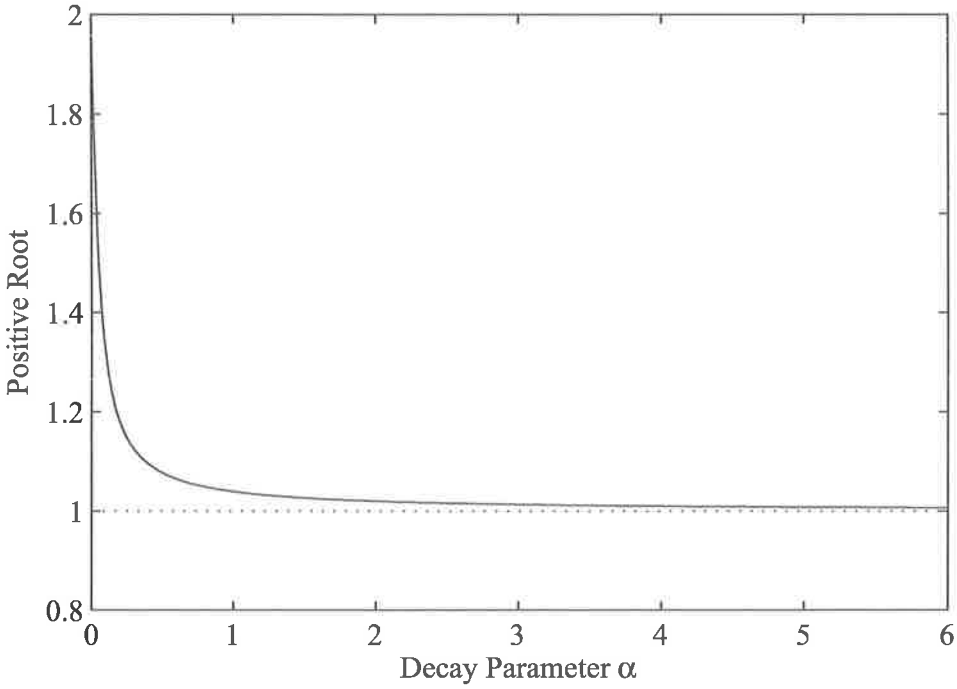


Figure 4.1: Positive roots for Example 3.1 as α varies

Proof. Case (i) GBM, we let $z = r + \alpha$ and $\delta = r - \nu$

$$\begin{aligned}
\lim_{\alpha\to\infty} k &= \lim_{z\to\infty} \frac{1}{\sigma}\left[-\left(\frac{z-\delta}{\sigma}-\frac{\sigma}{2}\right)+\sqrt{\left(\frac{z-\delta}{\sigma}-\frac{\sigma}{2}\right)^2+2z}\right] \\
&= \lim_{z\to\infty} \frac{\frac{1}{\sigma}\left[-\left(\frac{z-\delta}{\sigma}-\frac{\sigma}{2}\right)^2+\left(\frac{z-\delta}{\sigma}-\frac{\sigma}{2}\right)^2+2z\right]}{\frac{z-\delta}{\sigma}-\frac{\sigma}{2}+\sqrt{\left(\frac{z-\delta}{\sigma}-\frac{\sigma}{2}\right)^2+2z}} \\
&= \lim_{z\to\infty} \frac{\frac{2z}{\sigma}}{\frac{z-\delta}{\sigma}-\frac{\sigma}{2}+\sqrt{\left(\frac{z-\delta}{\sigma}-\frac{\sigma}{2}\right)^2+2z}} = 1.
\end{aligned}$$

Case (ii) JDP, by Corollary 3.4, we have

$$\Lambda_1(\nu+\alpha, r+\alpha) \leq \Lambda_2(\nu+\alpha, r+\alpha) \leq \Lambda_1(\nu+\alpha-\lambda\phi, r+\alpha).$$

By Case (i), we have $\Lambda_1(\nu+\alpha, r+\alpha)$ and $\Lambda_1(\nu+\alpha-\lambda\phi, r+\alpha)$ converging to one as $\alpha \to \infty$. Thus by the sandwich theorem we know that $\Lambda_2(\nu+\alpha, r+\alpha)$ must also converge to one. □

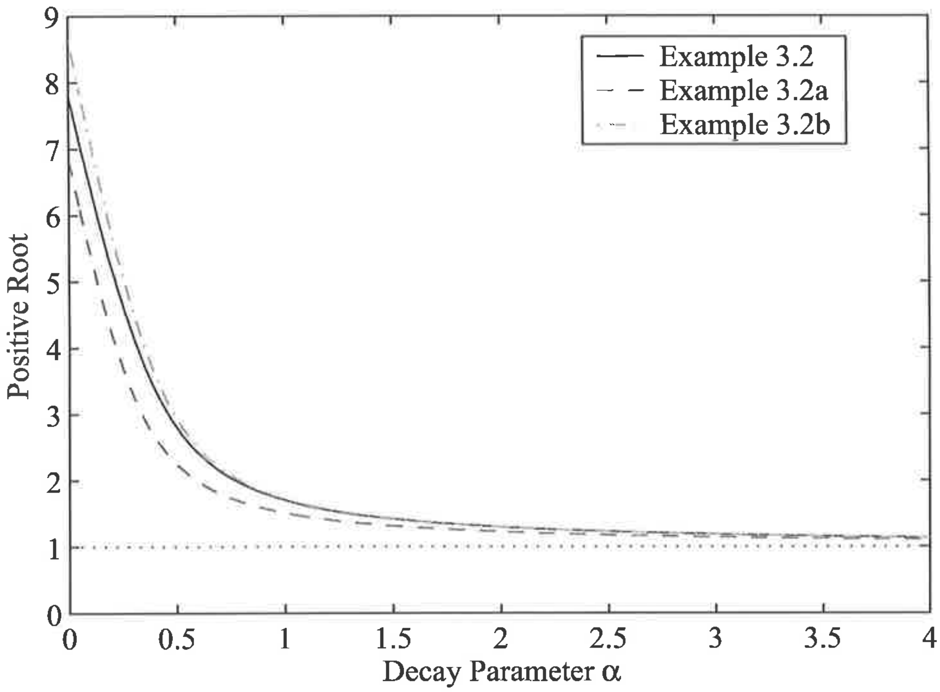


Figure 4.2: Positive roots for Example 3.2 as α varies

4.2.3 The Investment Trigger

Figure 4.3 shows the investment triggers for Example 3.1 as α varies on (0,0.5). The graph is approximately linear, $V^* \approx 25\alpha + 1.5$. In general (using order notation),

$$V^* = \frac{I}{\delta}\left[\left(r + \frac{1}{2}\sigma^2\right) + \alpha\right] + O\left(\frac{1}{\alpha}\right),$$

(see Lemma 4.4) and the related linear approximation gives investors a simple rule for making decisions when α is large. Figure 4.4 shows the investment triggers for Example 3.2, 3.2a and 3.2b as α varies on (0,2). Example 3.2a converges to linear $f(\alpha)$ and Examples 3.2 and 3.2b converge to linear $g(\alpha)$.

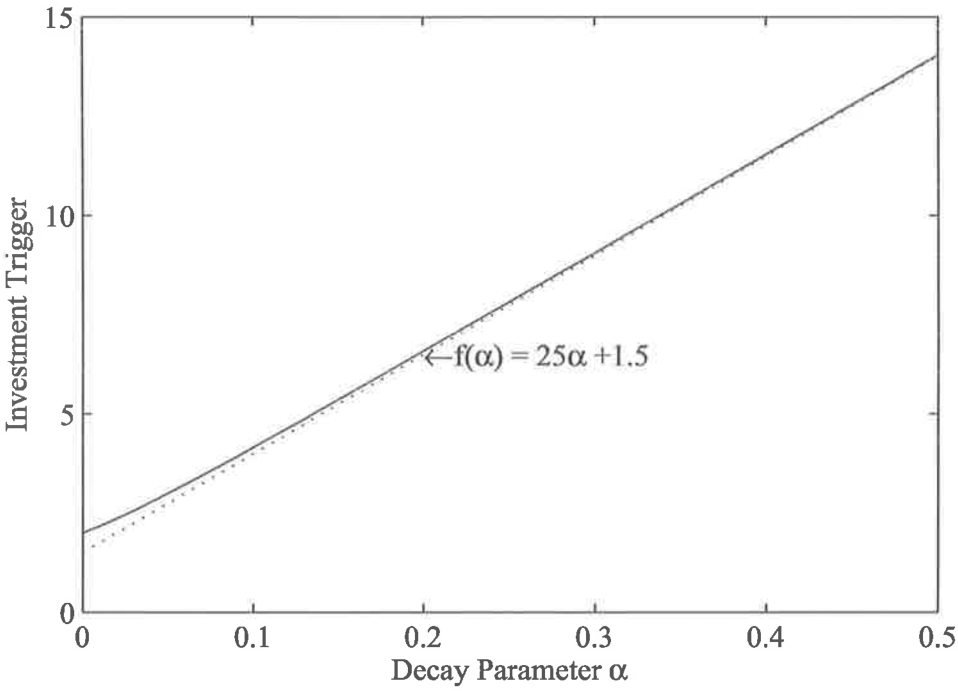


Figure 4.3: Investment triggers for Example 3.1 as α varies

Lemma 4.4. *If* $k = \Lambda_1(r+\alpha, r+\alpha)$, $V^* = \frac{kI}{(k-1)}$ *and* $\delta = r - \nu$ *then*

$$\lim_{\alpha\to\infty} \frac{V^*}{\alpha} = \frac{I}{\delta}, \tag{4.3}$$

$$\lim_{\alpha\to\infty} V^* - \frac{\alpha I}{\delta} = \frac{I}{\delta}(r + \sigma^2/2), \tag{4.4}$$

$$\lim_{\alpha\to\infty} \left(V^* - \frac{\alpha I}{\delta} - \frac{rI}{\delta} - \frac{\sigma^2 I}{2\delta}\right)\alpha = \frac{\sigma^2 I}{2}. \tag{4.5}$$

Proof. We need to establish an intermediate result:

$$\lim_{\alpha\to\infty} (k-1)(r+\alpha) = \delta. \tag{4.6}$$

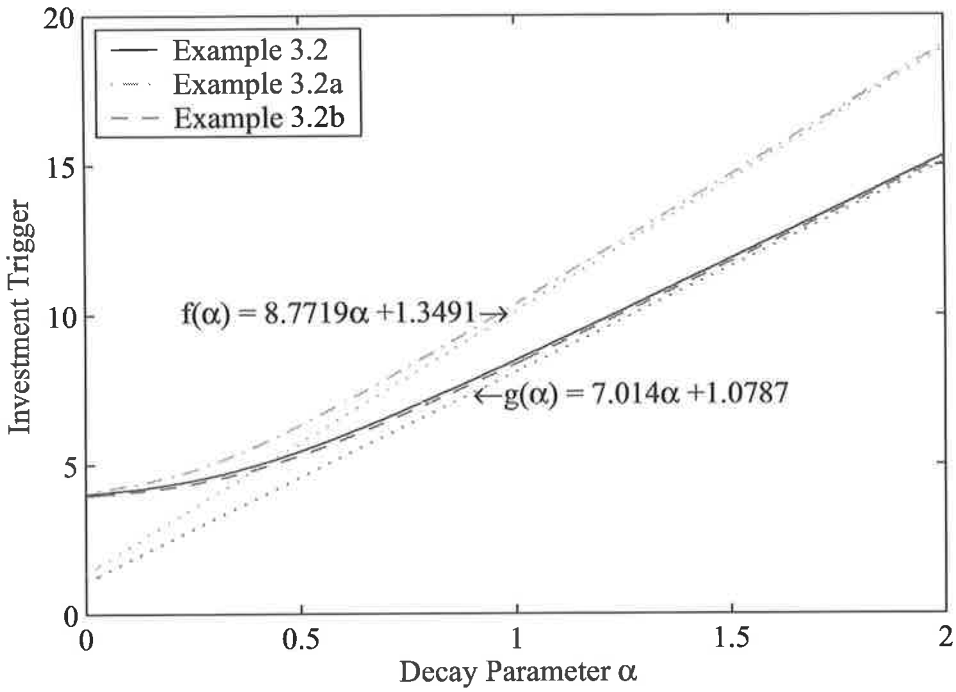


Figure 4.4: Investment triggers for Example 3.2 as α varies

Let $z = r + \alpha$,

$$\begin{aligned}
\lim_{\alpha\to\infty}(k-1)z &= \lim_{z\to\infty}\left(\frac{1}{\sigma}\left[-\left(\frac{z-\delta}{\sigma}-\frac{\sigma}{2}\right)+\sqrt{\left(\frac{z-\delta}{\sigma}-\frac{\sigma}{2}\right)^2+2z}\right]-1\right)z \\
&= \lim_{z\to\infty}\frac{z}{\sigma}\left[-\left(\frac{z-\delta}{\sigma}+\frac{\sigma}{2}\right)+\sqrt{\left(\frac{z-\delta}{\sigma}+\frac{\sigma}{2}\right)^2+2\delta}\right] \\
&= \lim_{z\to\infty}\frac{\frac{z}{\sigma}\left[-\left(\frac{z-\delta}{\sigma}+\frac{\sigma}{2}\right)^2+\left(\frac{z-\delta}{\sigma}+\frac{\sigma}{2}\right)^2+2\delta\right]}{\frac{z-\delta}{\sigma}+\frac{\sigma}{2}+\sqrt{\left(\frac{z-\delta}{\sigma}+\frac{\sigma}{2}\right)^2+2\delta}} \\
&= \lim_{z\to\infty}\frac{\frac{2z\delta}{\sigma}}{\frac{z-\delta}{\sigma}+\frac{\sigma}{2}+\sqrt{\left(\frac{z-\delta}{\sigma}+\frac{\sigma}{2}\right)^2+2\delta}} = \delta.
\end{aligned}$$

Combining Lemma 4.3 and (4.6) leads to

$$\lim_{\alpha\to\infty}\frac{V^*}{z} = \lim_{z\to\infty}\frac{kI}{z(k-1)} = \frac{I}{\delta}.$$

Subtracting $\frac{zI}{\delta}$ from V^* leads to

$$\begin{aligned} V^* - \frac{zI}{\delta} &= \frac{zI}{\delta}\left[\frac{\mu+\sigma+\sqrt{\mu^2+2z}}{\mu+\sqrt{\mu^2+2z}} - 1\right] \\ &= \frac{zI}{\delta}\left[\frac{\sigma}{\mu+\sqrt{\mu^2+2z}}\right] \\ &= \frac{I}{\delta}\left[\frac{z\sigma}{\mu+\sqrt{\mu^2+2z}}\right] \\ &= \frac{I}{\delta}\left[\frac{\sigma^2}{2}k\right]. \end{aligned}$$

By Lemma 4.3

$$\lim_{\alpha\to\infty} V^* - \frac{zI}{\delta} = \frac{\sigma^2 I}{2\delta}.$$

Subtracting the constant term from $V^* - \frac{zI}{\delta}$ we get:

$$V^* - \frac{zI}{\delta} - \frac{\sigma^2 I}{2\delta} = \frac{\sigma^2 I}{2\delta}(k-1).$$

By (4.6) we get

$$\lim_{z\to\infty}\left(V^* - \frac{zI}{\delta} - \frac{\sigma^2 I}{2\delta}\right)z = \frac{\sigma^2 I}{2\delta}\delta = \frac{\sigma^2 I}{2}.$$

□

4.2.4 The Investment Value

Figure 4.5 shows the investment values for Example 3.1 as α varies on (0,30). Figure 4.6 shows the investment values for Example 3.2 as α varies on (0,1000). These graphs suggest that the investment value is converging to the initial value as α increases. This result is not particularly surprising because we would expect that $I(t)$ will drop to zero and the investment will be made almost instantly when α is large. We note that

$$\max_{\tau} E[V(\tau)e^{-r\tau}|V(0) = V] = V.$$

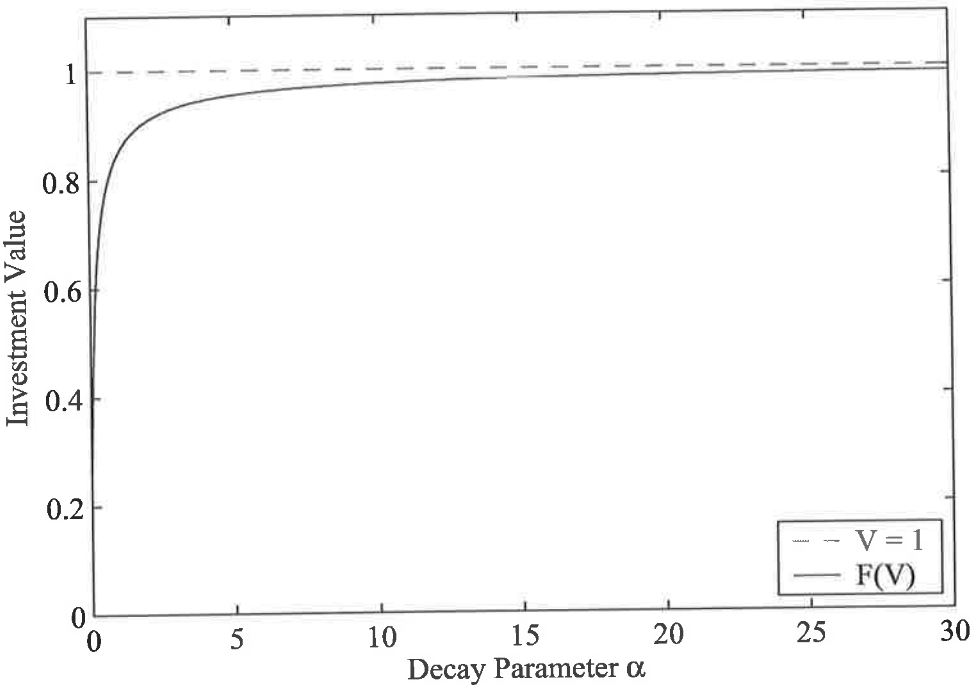


Figure 4.5: Investment values for Example 3.1 as α varies

Proposition 4.5. *The investment value, $F(V)$ defined in (4.2), converges to the initial value, V, as the decay parameter α increases.*

Proof. We apply Lemma 4.3 twice to get the result.

$$\begin{aligned}
\lim_{\alpha\to\infty} F(V) &= \lim_{\alpha\to\infty} (V^* - I)\left(\frac{V}{V^*}\right)^k \\
&= \lim_{\alpha\to\infty} (V^* - I)\left(\frac{V}{V^*}\right) \\
&= \lim_{\alpha\to\infty} V\left(\frac{V^* - I}{V^*}\right) \\
&= \lim_{\alpha\to\infty} V\left(\frac{\frac{kI}{k-1} - I}{\frac{kI}{k-1}}\right) \\
&= \lim_{\alpha\to\infty} V\left(\frac{I}{kI}\right) \\
&= V.
\end{aligned}$$

□

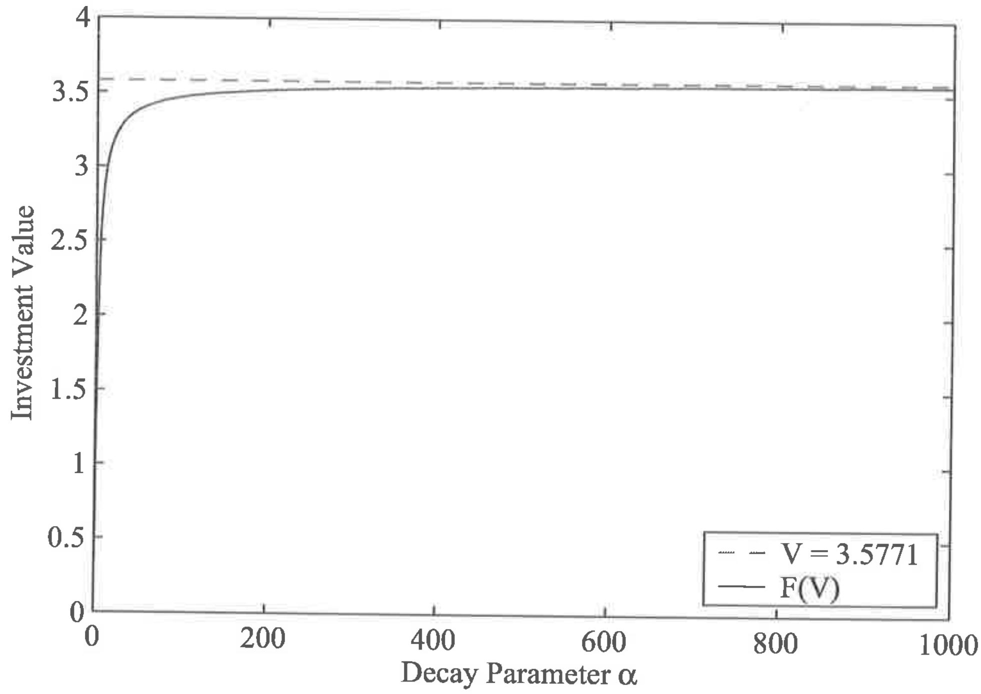


Figure 4.6: Investment values for Example 3.2 as α varies

4.2.5 Convergence of finite models

In Section 3.4, the binomial model was used to measure how quickly the finite-model converged to the perpetual model. We are also interested in the rate of convergence for the decreasing cost models. Algorithm 2.7 is easily extended to decreasing strike prices $K(t) = Ke^{-\alpha t}$ by replacing (2.10) and (2.11) with

$$\begin{aligned} V(j,N) &= (S(j,N) - Ke^{-\alpha T})^+, \\ V(j,n) &= \max(E(j,n), (S(j,n) - Ke^{-\alpha \frac{nT}{N}})^+), \end{aligned}$$

where the N is the number of steps in the binomial model. Figure 4.7 shows the rate of convergence for Example 3.1 with $N = 5T$ and decay parameters $\alpha \in [0, 0.462, 0.693, 1.099]$. The graph suggests that the finite-time model converges more rapidly as the decay parameter increases.

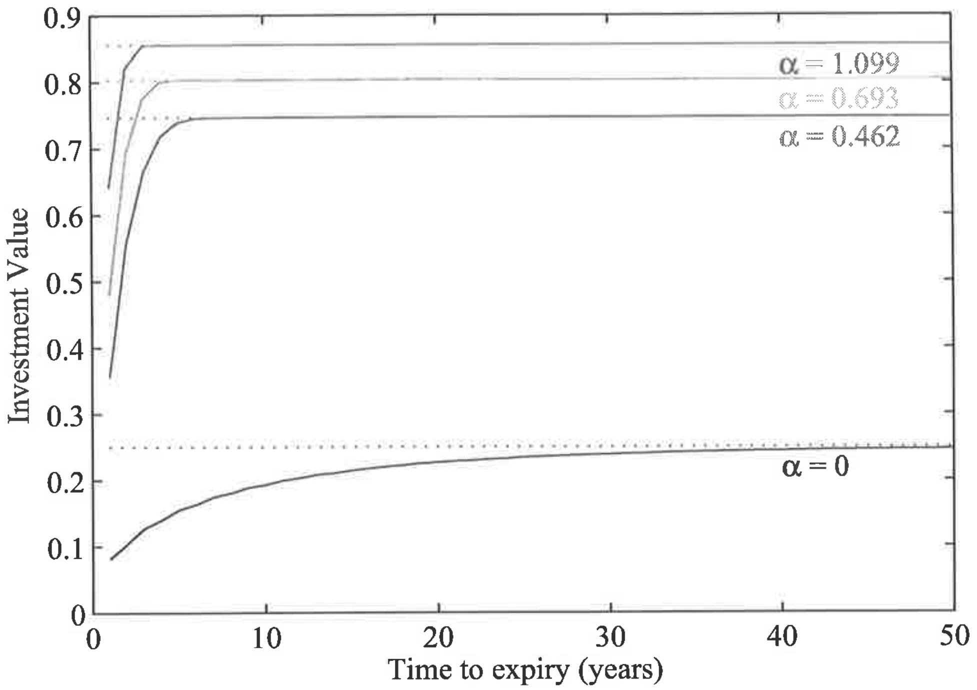


Figure 4.7: The convergence of decreasing costs models

4.2.6 Stopping Times

The stopping time formulae presented in Section 3.4 can be extended to the decreasing cost model by replacing ν with $\nu + \alpha$ and r with $r + \alpha$. Figure 4.8 shows the stopping time probabilities for Example 3.1. In the base case ($\alpha = 0$), the drift is negative ($\mu = -0.02$) and so there is only a 50% chance that the investment will be made. The investment is guaranteed for positive drift (i.e. when $\alpha > 0.02$). Larger decay parameters are needed to ensure that the investment occurs before time $t = 100$ (i.e. $\alpha > 0.1$). This probability is verified at selected points using the simulation estimate (3.19). The confidence intervals for $N = 1000$ runs are shown as vertical lines.

Figure 4.9 shows some stopping time probabilities for Example 3.2. Since Example 3.2 is a JDP model we do not have an exact solution for the stopping time probability. In this figure, the graphs connect the sample means for simulation es-

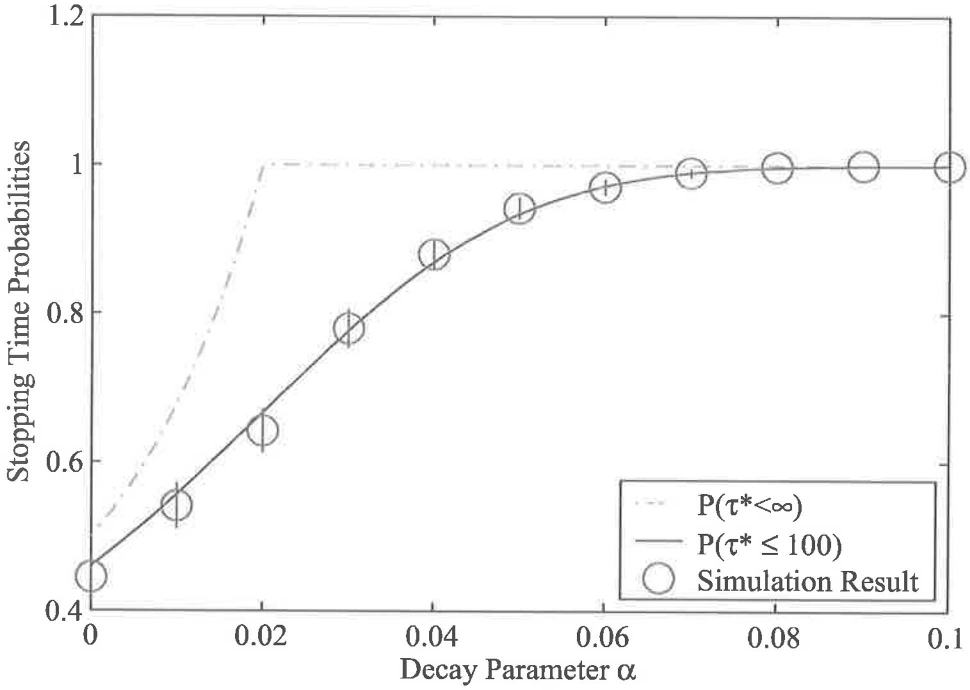


Figure 4.8: Stopping time probabilities for Example 3.1 as α varies

timate (3.19), with $N = 1000$ runs and $t = 100$ or $t = 1000$, at selected values of α. Combining these results with those in Figure 4.8, we note that the stopping time probabilities converge to one as the decay parameter increases.

Figure 4.10 shows the risk-neutral expected stopping times $E[\tau^*]$ and $E[\tau^*|\tau^* \leq 100]$ for Example 3.1. The expected stopping times are infinite when $\alpha \leq 0.02$. At other times, the expected stopping time decreases (i.e. on average the investment will be made sooner) when the decay parameter increases. The expected stopping times and the conditional expected stopping times are the same for $\alpha > 0.1$, because the investment will always be made before $t = 100$. This conditional expected stopping time is verified at selected points using the simulation estimate (3.20). The confidence intervals for $N = 1000$ runs are shown as vertical lines.

Figure 4.11 shows some conditional expected stopping times for Example 3.2. Since Example 3.2 is a JDP model we do not have an exact solution for the con-

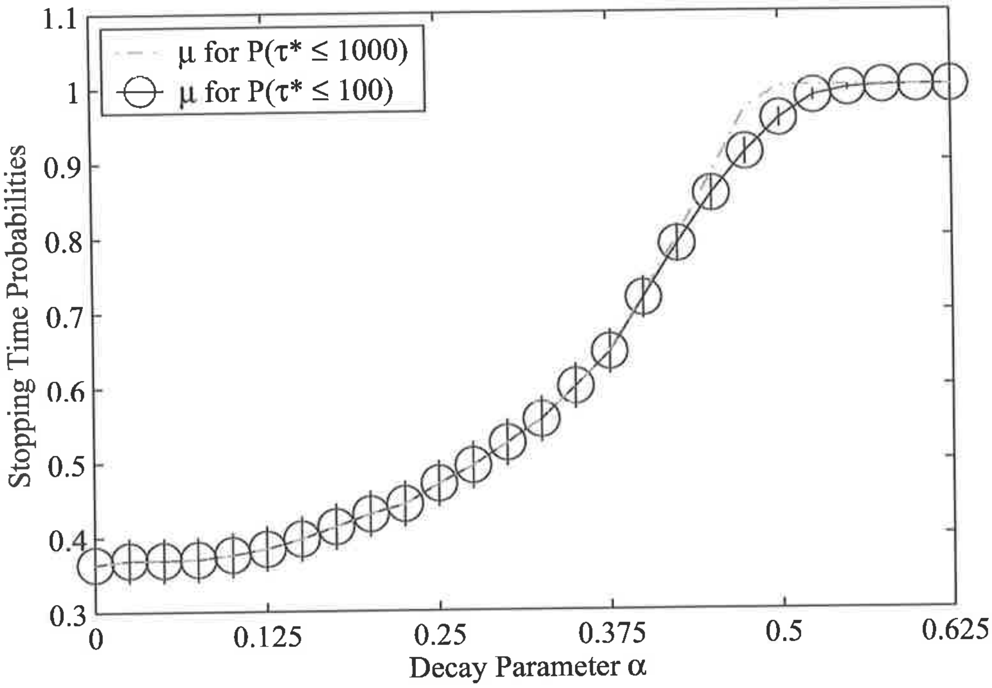


Figure 4.9: Stopping time probabilities for Example 3.2 as α varies

ditional expected stopping times. In this figure, the graphs connect the sample means for the simulation estimate (3.20), with $N = 1000$ runs and time $t = 100$ or $t = 1000$, at selected values of α. Since simulation data for $t = 100$ and $t = 1000$ are collected from the same sample paths, the sample means for $E[\tau^*|\tau^* \leq 100]$ are never greater than than those for $E[\tau^*|\tau^* \leq 1000]$. However, this property may not hold when different sample paths are used. In fact, we observed this anomaly in some initial simulations where the processes were instantiated with the same seed but run longer (for the case $t = 1000$) causing the sample paths to diverge. Combining these results with those in Figure 4.10, we note that the conditional expected stopping times decrease as the decay parameter increases.

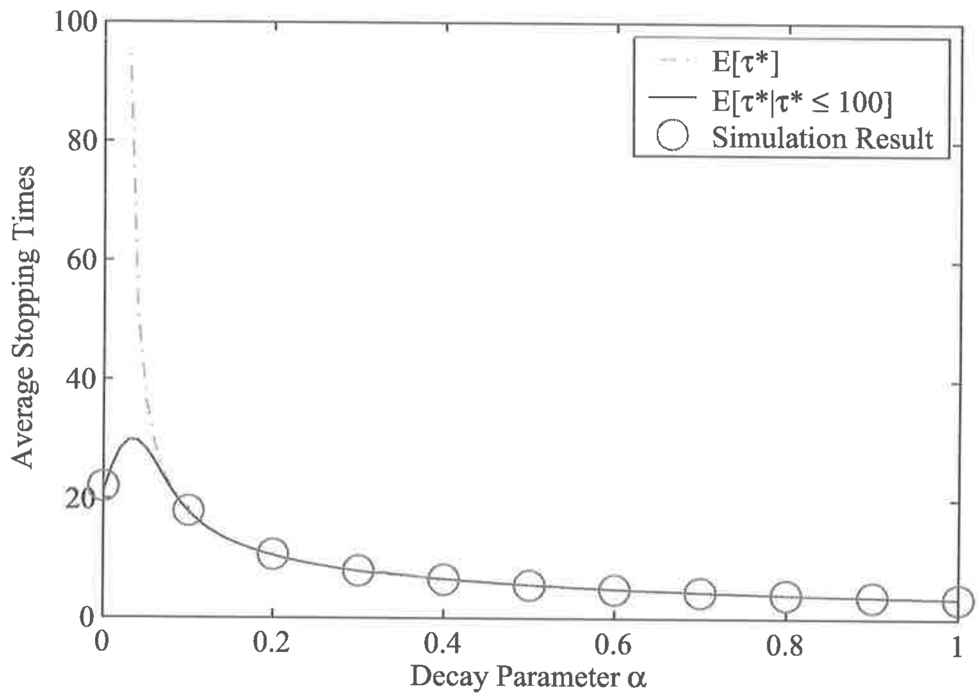


Figure 4.10: Expected stopping times for Example 3.1 as α varies

4.3 Errors in the decay parameter

In the previous section, we presented a model which gives the optimal strategy for a given decay parameter. However, in Section 4.1 we cited two cases where decay predictions had been optimistic (i.e. the predicted decay parameter was greater than the true decay parameter). Moore's original estimate for decay was roughly 50% greater than his revised estimate. Similarly, Gilder's decay estimate was about 50% greater than Odlyzko's estimate. Section 4.3.1 develops the cost error model and Section 4.3.2 applies various error scenarios to Example 3.1.

4.3.1 The Cost Error Model

Given the decay parameter α, the analysis in Section 4.2 tells us to invest at time

$$\tau^* = \inf\{t > 0 | V(t) \geq e^{-\alpha t} V^*\},$$

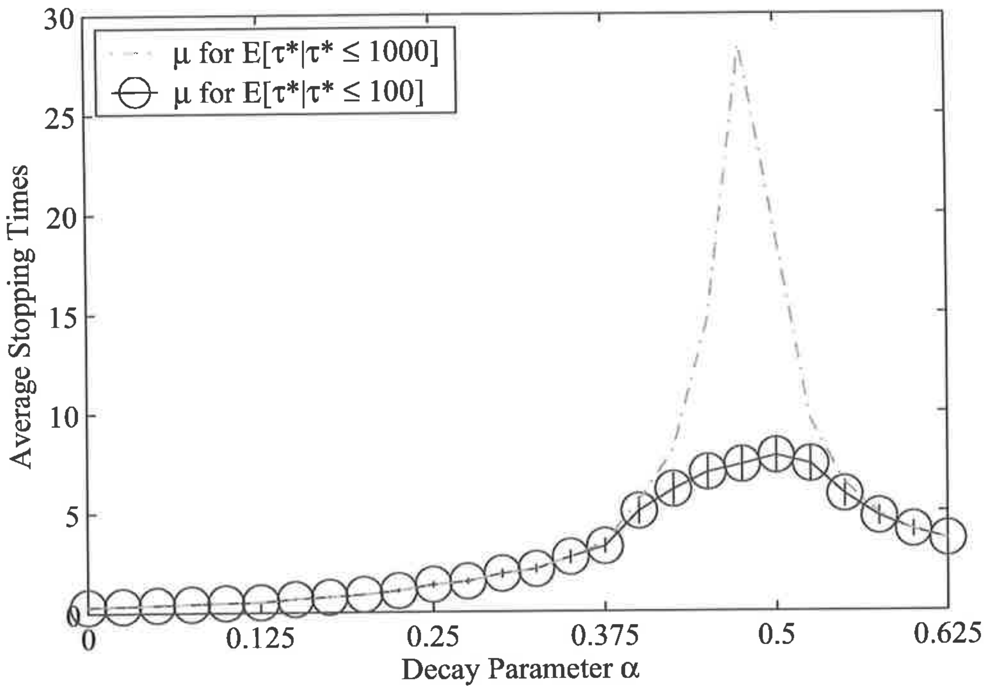


Figure 4.11: Conditional expected stopping times for Example 3.2 as α varies

with

$$\begin{aligned} k &= \Lambda(\nu+\alpha, r+\alpha), \\ V^* &= \frac{k}{k-1}I. \end{aligned}$$

This strategy yields the maximum possible value

$$F(V) = \begin{cases} (V^*-I)\left(\frac{V}{V^*}\right)^k, & 0 < V < V^*; \\ (V-I), & V \geq V^*. \end{cases}$$

Now suppose that the predicted decay parameter is α_1, then we believe that the optimal time is to invest is

$$\tau_1^* = \inf\{t > 0 | V(t) \geq e^{-\alpha_1 t} V_1^*\},$$

with

$$\begin{aligned} k_1 &= \Lambda(\nu+\alpha_1, r+\alpha_1), \\ V_1^* &= \frac{k_1}{k_1-1} I. \end{aligned}$$

Following this strategy, the predicted investment value is

$$F_1(V) = \begin{cases} (V_1^* - I)\left(\frac{V}{V_1^*}\right)^{k_1}, & 0 < V < V_1^*; \\ (V - I), & V \geq V_1^*. \end{cases}$$

Summarizing these results, the predicted value is $F_1(V)$ and the maximum value is $F(V)$. In reality, neither of these values will be achieved because the errors in the estimate will cause the investment to be made at the wrong time: the analysis recommends investment at time τ_1^* but the optimal investment time is τ^*. Since this strategy is not optimal, the expected value will be lower than the optimal value $F(V)$. The suboptimal value, $H(V)$, is given by

$$\begin{aligned} H(V) &= E_0[(V(\tau_1^*) - Ie^{-\alpha\tau_1^*})e^{-r\tau_1^*} I(\tau_1 < \infty)] \\ &= V_1^* E_0[e^{-(r+\alpha_1)\tau_1^*} I(\tau_1 < \infty)] \\ &\quad - I E_0[e^{-(r+\alpha)\tau_1^*} I(\tau_1 < \infty)] \\ &= V_1^* \left(\frac{V}{V_1^*}\right)^{k_1} - I\left(\frac{V}{V_1^*}\right)^{k_2}, \end{aligned}$$

where

$$k_2 = \Lambda(\nu+\alpha_1, r+\alpha).$$

Unlike the original expression for $F(V)$, the expression for $H(V)$ does not involve the positive part $(\cdot)^+$ because the investment strategy is based on error-prone data and may yield a negative net payoff. However, we will see that this does not happen in the current example.

4.3.2 Numerical Examples

Gilder's decay estimate was approximately 50% greater than that given by Odlyzko. Figure 4.12 shows the investment values and expected stopping times when the

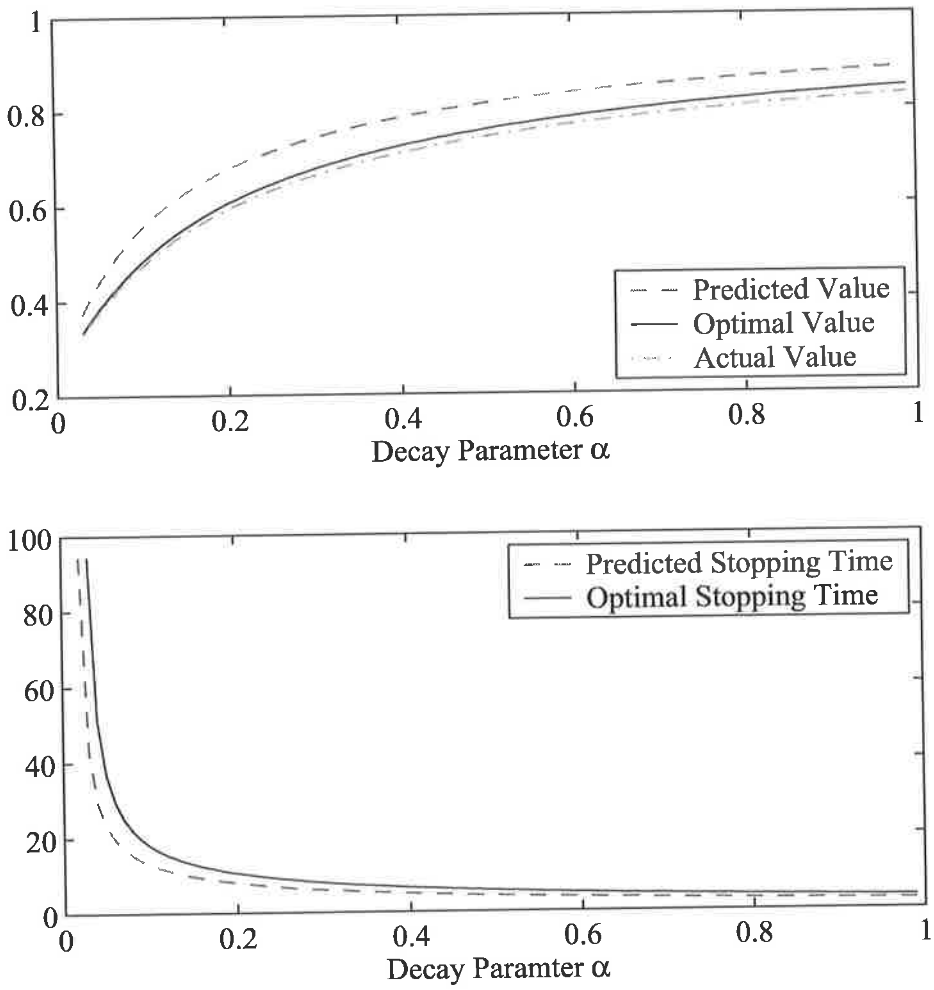


Figure 4.12: Optimistic decay predictions

predicted decay parameter is 50% greater than the true decay parameter. These graphs suggest that optimistic predictions will encourage an investor to invest earlier and that the investor will expect unreasonably high net investment values. The premature investment will also cause the investor to forfeit some net investment value, although the reduction in value is less than 5%. In this example, we have

V = I = 1, so

$$H(1) = \left(\frac{1}{V_1^*}\right)^{k_2}\left(\left(\frac{1}{V_1^*}\right)^{k_1-1-k_2} - 1\right)$$

and $k_1 - 1 - k_2 < 0$; hence the actual value is greater than zero. However, this condition may not hold when $V < I$ (which is likely for many start-up companies), and we provide an example with $H(V) < 0$ at the end of this section. Thus optimistic predictions could lead to negative net payoffs.

Proof. Want to show $k_1 - k_2 - 1 < 0$. For GBM $V(t)$,

$$\begin{aligned} k_1 - k_2 &= \frac{1}{\sigma}[-\mu_1 + \sqrt{\mu_1^2 + 2(r+\alpha_1)}] - \frac{1}{\sigma}[-\mu_1 + \sqrt{\mu_1^2 + 2(r+\alpha)}] \\ &= \frac{1}{\sigma}[\sqrt{\mu_1^2 + 2(r+\alpha_1)} - \sqrt{\mu_1^2 + 2(r+\alpha)}] \\ &= \frac{1}{\sigma}\frac{2(\alpha_1 - \alpha)}{\sqrt{\mu_1^2 + 2(r+\alpha_1)} + \sqrt{\mu_1^2 + 2(r+\alpha)}}. \end{aligned} \tag{4.7}$$

For the conservative case ($\alpha \geq \alpha_1$), (4.7) is non-positive so $k_1 - k_2 < 0 < 1$. We want to show that the relationship also holds for $\alpha_1 > \alpha > 0$.

In our example,

$$\begin{aligned} k_1 - k_2 &= \frac{1}{2}[\sqrt{2500\alpha_1^2 + 100\alpha_1 + 9} - \sqrt{2500\alpha_1^2 - 100\alpha_1 + 200\alpha + 9}] \\ &= \frac{1}{2}\frac{2500\alpha_1^2 + 100\alpha_1 + 9 - (2500\alpha_1^2 - 100\alpha_1 + 200\alpha + 9)}{\sqrt{2500\alpha_1^2 + 100\alpha_1 + 9} + \sqrt{2500\alpha_1^2 - 100\alpha_1 + 200\alpha + 9}} \\ &= \frac{100(\alpha_1 - \alpha)}{\sqrt{2500\alpha_1^2 + 100\alpha_1 + 9} + \sqrt{2500\alpha_1^2 - 100\alpha_1 + 200\alpha + 9}}. \end{aligned}$$

Set

$$g(\alpha) = 100(\alpha_1 - \alpha) - \sqrt{2500\alpha_1^2 + 100\alpha_1 + 9} - \sqrt{2500\alpha_1^2 - 100\alpha_1 + 200\alpha + 9},$$

then $k_1 - k_2 - 1 < 0$ iff $g(\alpha) < 0$. We want to show that $g(\alpha) < 0$ for all $\alpha \geq 0$, thus we need to show that the following conditions hold:

$$g(0) < 0, \tag{4.8}$$

$$g'(\alpha) \leq 0. \tag{4.9}$$

$$\begin{aligned}
g(0) &= 100(\alpha_1) - \sqrt{2500\alpha_1^2 + 100\alpha_1 + 9} - \sqrt{2500\alpha_1^2 - 100\alpha_1 + 9} \\
&< 100(\alpha_1) - \sqrt{2500\alpha_1^2 + 100\alpha_1 + 1} - \sqrt{2500\alpha_1^2 - 100\alpha_1 + 1} \\
&= 100(\alpha_1) - \sqrt{(50\alpha_1 + 1)^2} - \sqrt{(50\alpha_1 - 1)^2} \\
&= 100(\alpha_1) - 50\alpha_1 + 1 - |50\alpha - 1| \\
&\leq 50\alpha_1 - 1 - (50\alpha - 1) \\
&= 0.
\end{aligned}$$

Since the first condition holds, we now need to show that $g'(\alpha) \leq 0$:

$$\begin{aligned}
g'(\alpha) &= -100 - \frac{1}{2}\frac{200}{\sqrt{2500\alpha_1^2 - 100\alpha_1 + 200\alpha + 9}} \\
&< 0.
\end{aligned}$$

Hence the second condition also holds and we have established that $g(\alpha) < 0$. Thus we have shown that $k_1 - k_2 - 1 < 0$. □

It is also useful to consider pessimistic predictions. The dominance of optimistic estimates may lead a cautious investor to use a lower value than the true decay parameter. Figure 4.13 shows the investment value and expected stopping time when the estimated decay parameter is 50% less than the true decay parameter. These graphs show that pessimistic predictions will lead an investor to wait too long for investment and thereby reduce the investment value. Again the reduction in value is less than 5%. For pessimistic predictions, $\alpha > \alpha_1$ and $k_2 > k_1$, so we have

$$\begin{aligned}
H(V) &= V_1^* \left(\frac{V}{V_1^*}\right)^{k_1} - I\left(\frac{V}{V_1^*}\right)^{k_2} \\
&> V_1^* \left(\frac{V}{V_1^*}\right)^{k_1} - I\left(\frac{V}{V_1^*}\right)^{k_1} \\
&= F_1(V) \geq 0.
\end{aligned}$$

Thus the actual value is bounded by the predicted value. This means that actual value is always greater than zero and so the net payoff is never negative. Therefore pessimism is a safe strategy.

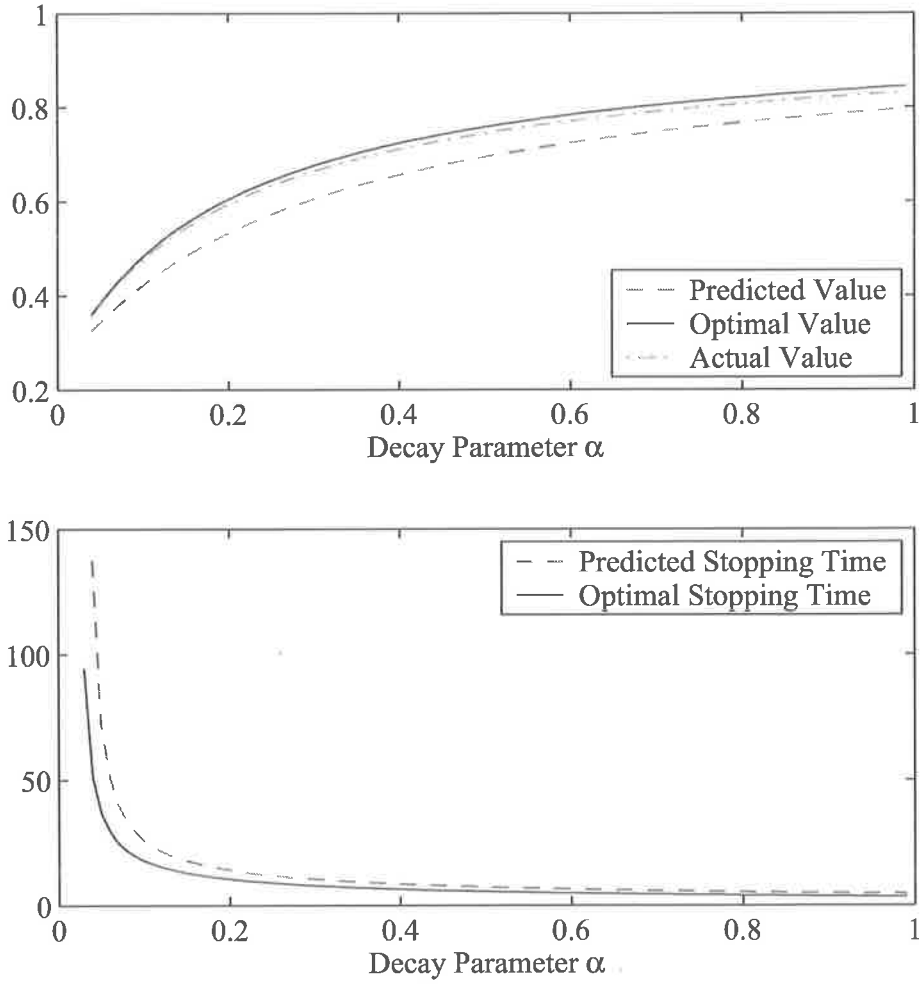


Figure 4.13: Pessimistic decay predictions

The above examples suggest that our model is robust; it is insensitive to relatively large errors (±50%). In Figures 4.14 and 4.15, the decay parameter is fixed at $\alpha = 0.5$ and the relative error varies from -400% to 400%. Figure 4.14 shows the investment values and Figure 4.15 shows the relative difference and loss. These graphs demonstrate the need for measuring the decay parameter. Extremely large errors (e.g. 400%) may reduce the value by as much as 30%. Furthermore, in some special cases (e.g. $V = 0.1I$), the net payoffs may even be negative (Figure 4.16).

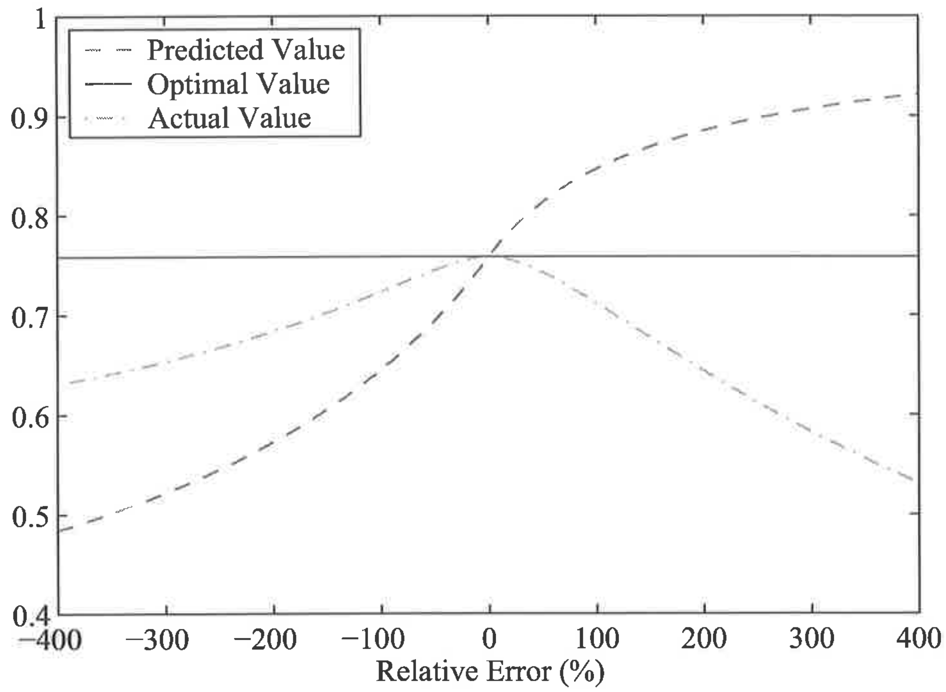


Figure 4.14: Investment value versus the relative error

4.4 Errors in the traffic growth parameter

In the previous sections we extended the fixed cost model to support decreasing costs and studied the impact of errors in the decay parameter. In this section we shall explain how to increase the traffic rate and investigate errors in the traffic parameter. For simplicity we shall work with the fixed cost model described in Corollary 4.1. However, we note that this model can be easily transformed into a decreasing cost model by adding the decay parameter to the drift and interest rate.

In Chapter 3, we explained that the traffic rate is included in the drift ν and so

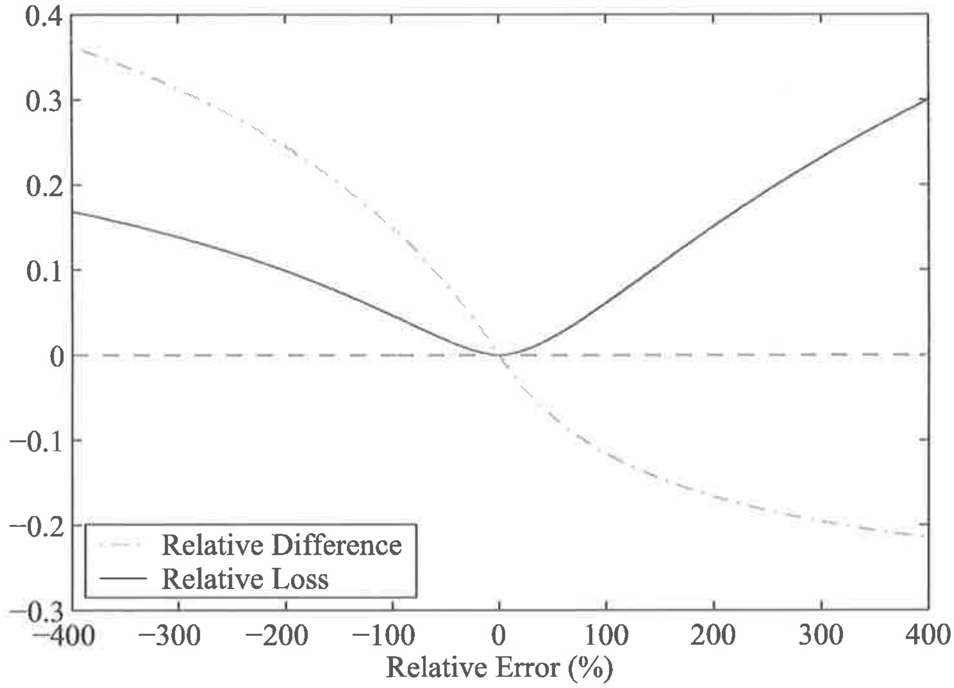


Figure 4.15: Relative difference and loss versus the relative error

increasing the traffic rate by traffic parameter γ is simply a matter of adding γ to the drift ν. This means that the positive root is $k = \Lambda(\nu + \gamma, r)$. We note that this is different from the decreasing cost model with decay parameter γ. In that case, the positive root is $k = \Lambda(\nu + \gamma, r + \gamma)$. In Chapter 3, we noted that there is no optimal trigger in the GBM model when the drift is greater than or equal to the interest rate, and that there is a similar relationship for the JDP model. Since both ν and γ are increased by the same amount in the decreasing cost model, there is an optimal trigger in decreasing cost model whenever an optimal trigger exists in the original model. However, only the drift parameter is increased when the traffic rate is increased and so there is no optimal trigger in the GBM model when $\nu + \gamma \geq r$.

Suppose that the estimated traffic parameter is wrong and that the true traffic parameter is γ, then the optimal time to invest is

$$\tau^* = \inf\{t > 0 | V(t)e^{\gamma t} \geq V^*\},$$

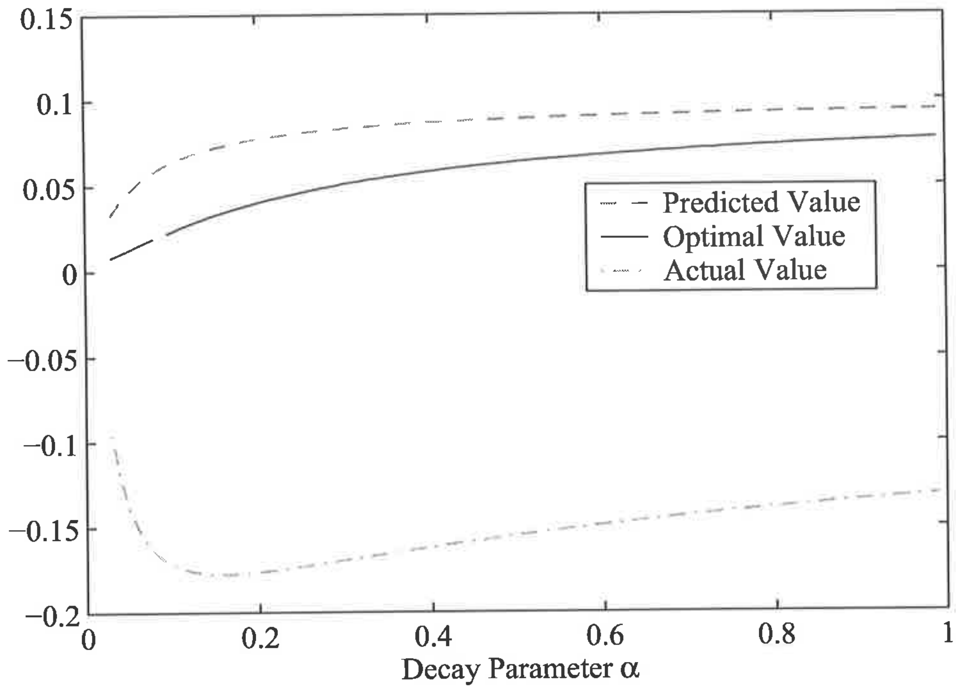


Figure 4.16: Optimistic decay predictions leading to negative investment values

with

$$\begin{aligned} k &= \Lambda(\nu+\gamma, r), \\ V^* &= \frac{k}{k-1} I. \end{aligned}$$

This strategy yields the maximum possible value

$$F(V) = \begin{cases} (V^* - I)\left(\frac{V}{V^*}\right)^k, & 0 < V < V^*; \\ (V - I), & V \geq V^*. \end{cases}$$

However, the investor believes that the traffic parameter is γ_1 and so it would appear that the optimal investment time is

$$\tau_1^* = \inf\{t > 0 | V(t)e^{\gamma_1 t} \geq V_1^*\},$$

with

$$\begin{aligned} k_1 &= \Lambda(\nu + \gamma_1, r), \\ V_1^* &= \frac{k_1}{k_1 - 1} I. \end{aligned}$$

Following this strategy the investor expects to receive

$$F_1(V) = \begin{cases} (V_1^* - I)\left(\frac{V}{V_1^*}\right)^{k_1}, & 0 < V < V_1^*; \\ (V - I), & V \geq V_1^*. \end{cases}$$

Based on the analysis, the investor will expect to receive $F_1(V)$, however the most they can receive is $F(V)$. In reality, neither of these values will be achieved because the errors in the estimate will lead to investment being made at the wrong time: the analysis recommends investment at time τ_1^* but the optimal investment time is τ^*. Since the investor observes $V(t)e^{\gamma t}$ instead of $V(t)e^{\gamma_1 t}$, the investment will be made at another time

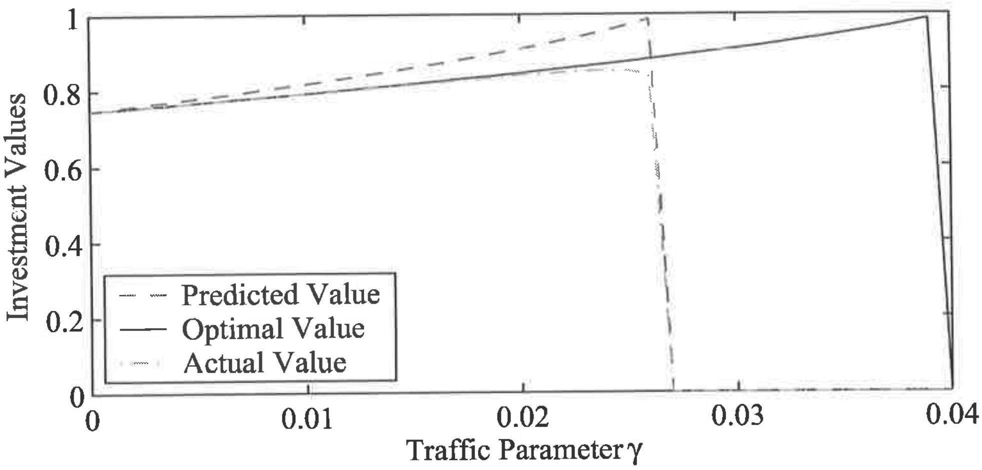
$$\tau_2^* = \inf\{t > 0 | V(t)e^{\gamma t} \geq V_1^*\}.$$

Since this strategy is not optimal, the investment value will be less than the optimal value $F(V)$. The suboptimal value, $H(V)$, is given by

$$\begin{aligned} H(V) &= E_0[(V(\tau_2^*) - I)e^{-r\tau_2^*} I(\tau_1 < \infty)] \\ &= E_0[(V_1^* - I)e^{-r\tau_2^*} I(\tau_1 < \infty)] \\ &= (V_1^* - I)[e^{-r\tau_2^*} I(\tau_1 < \infty)] \\ &= (V_1^* - I)\left(\frac{V}{V_1^*}\right)^k. \end{aligned}$$

As in the predicted traffic model, the investment will only be made when the optimal trigger exists and $V_1^* > I$.

Figure 4.17 shows the investment values and expected stopping times for Example 3.1 with $\alpha = 0.462$, γ varying on $[0, 0.04]$ and the predicted traffic parameter γ_1 being 50% greater than the true traffic parameter γ. From earlier, we know that there is no optimal trigger when the drift is greater than the interest rate, and so



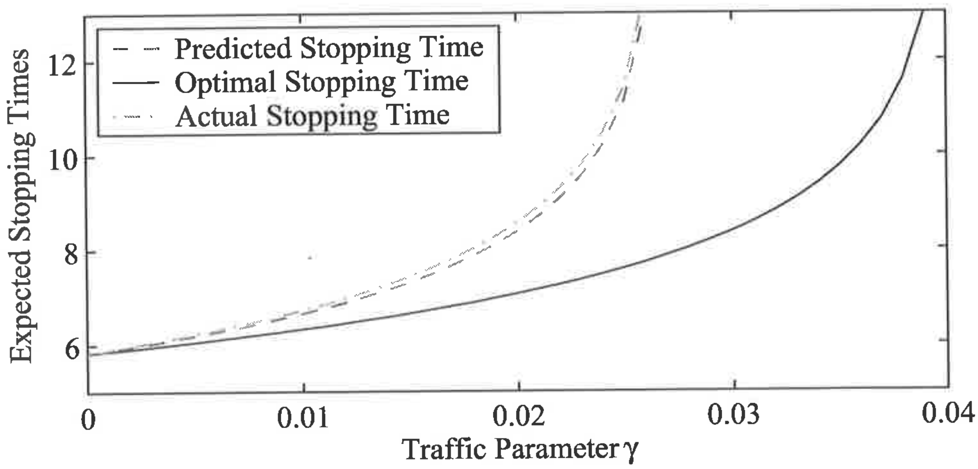


Figure 4.17: Optimistic traffic predictions

the predicted model and true model have no optimal trigger when γ is greater than $0.02\dot{6}$ and 0.04 respectively. Prior to $0.02\dot{6}$, the predicted model indicates much larger investment values and this analysis prompts the investor to delay investment until the value process reaches a higher threshold. After $0.02\dot{6}$, the predicted model indicates that there is no optimal trigger but this is false for $\gamma < 0.04$.

4.5 Conclusion

Two simple models for building new infrastructure were presented in the previous chapter. In this chapter we extended these models to support decreasing investment costs. We found that the investment values are close for a JDP model and a related GBM when the decay parameter is large. We then studied the impact of errors in the decay parameter. We found that they reduced the investment value, but for relatively large errors the reduction was less than 5%. For much larger errors the loss is more significant. These results suggest that the optimal strategy, when duly followed, will protect investors from large losses.

Unfortunately, models with small errors in decay parameter promise much higher investment values. In the light of these inflated values, the investor may disregard the optimal strategy and invest prematurely. Premature investment (based on optimistic predictions) may produce a negative investment value. Thus the investor should adopt a conservative practice of following the optimal strategy and treating predictions of the decay parameter with caution.

We also studied errors in the traffic rate and provided some brief analysis for the original GBM example. As with the cost error model, the traffic error model predicts incorrect investment values and encourages investors to invest at a suboptimal time. Unlike the cost error model, however, the traffic error model may indicate that there is no optimal trigger. This may lead an investor to postpone the investment indefinitely even though an optimal trigger may exist.

In the next two chapters we shall develop a strategy for deciding when to increase the transmission capacity on an existing link. We shall find an analytical solution for two demand processes: a geometric Brownian motion and a logistic process.

Chapter 5

Increasing Link Capacity

In the previous chapters we investigated the option of adding a link between two cities. This chapter considers the situation where the link has already been built and there is an option to increase capacity on the existing link. We present a general strategy for deciding when to make this investment and then provide an analytical solution for a GBM demand process.

5.1 Introduction

The previous models assume that no cash flows can be obtained before the investment is made (e.g. no link has been built) and that the transmission capacity is unlimited, that is

$$V(t) = E\left[\int_t^\infty P(s)e^{-r(s-t)}ds\right]$$

where $P(s)$ is the cash flow per unit time. D'Halluin, Forsyth and Vetzal [20] developed some PDEs for increasing the transmission capacity from a lower level (e.g. OC-3) to a higher level (e.g. OC-48) and then applied a numerical PDE solver. In this chapter we formulate a similar model and find an analytical solution.

Section 5.2 presents a general strategy for deciding when to increase the transmission capacity from level S_0 to level S_1. This strategy could be applied to various demand processes $D(t)$. We assume that regulators have capped the price so that each connection produces β dollars in revenue. The transmission capacity (also called the supply level), denoted by S, is the maximum number of connections which can be carried during a given time interval. If $D(t) \le S$, the revenue per unit time is $\beta D(t)$. If $D(t) > S$ some of the demand will not be met, packets will be dropped or otherwise prevented from using the network, and so the revenue per unit time is βS. Thus the revenue per unit time is $\beta \min(D(t), S)$.

In Section 5.3 we use a geometric Brownian motion to model the demand process. This assumption was employed in the model of d'Halluin, Forsyth and Vetzal [20], and in all of the previous models presented in this thesis. However, this assumption will be questioned in the following chapter.

5.2 A General Strategy

In this section we devise a general strategy for deciding when to increase the transmission capacity from S_0 to S_1. We first determine an expression for the expected investment value that will be received if the investment is made when the demand process $D(t)$ first hits the threshold y. This function is represented by $F(D, y)$ where D is the initial value for the demand process ($D(0) = D$). We then maximize this function over all $y > 0$ to find the optimal trigger

$$y^* = \operatorname*{argmax}_y F(D, y).$$

The investment will be made when $D(t)$ hits some threshold x. We define the stopping time $\tau(y) = \inf\{t \ge 0 : D(t) \ge y\}$ and write $\tau = \tau(y)$ when there is no confusion. The revenue per unit time will depend on whether or not the investment has been made:

- Prior to time τ, the transmission capacity is S_0. Thus the revenue per unit time is $\beta \min(D(t), S_0)$.
- After time τ, the transmission capacity is increased to S_1. Thus the revenue per unit time is $\beta \min(D(t), S_1)$.

The present value of all revenues is obtained by multiplying the revenue per unit time by e^{-rt} (where r is the risk-free interest rate) and integrating over $[0, \infty)$. As discussed in Chapter 4, we assume that investment costs are decreasing, thus $I(t) = Ie^{-\alpha t}$. For a sample path with stopping time τ, the present value of all revenues is

$$R(\tau) = \beta \int_0^\tau \min(D(t), S_0)e^{-rt}dt + \beta \int_\tau^\infty \min(D(t), S_1)e^{-rt}dt,$$

and the present value of the investment cost is

$$I(\tau) = Ie^{-\alpha\tau}e^{-r\tau} = e^{-(r+\alpha)\tau}.$$

Thus the expected investment value for threshold y is

$$F(D, y) = E\left[\int_0^\tau \beta \min(D(t), S_0)e^{-rt}dt + \int_\tau^\infty \beta \min(D(t), S_1)e^{-rt}dt - Ie^{-(r+\alpha)\tau}\right].$$

The optimal trigger is

$$y^* = \operatorname*{argmax}_y F(D, y),$$

and the optimal investment value is

$$F(D) = F(D, y^*).$$

The function $F(D, y)$ can be divided into three separate terms

$$\begin{aligned} F(D, y) &= \beta E\left[\int_0^\tau \min(D(t), S_0)e^{-rt}dt\right] + \beta E\left[\int_\tau^\infty \min(D(t), S_1)e^{-rt}dt\right] \\ &\quad -IE[e^{-(r+\alpha)\tau}]. \end{aligned}$$

In the remainder of this section, we shall present some general theorems which can be used to calculate these terms for various demand processes. The general strategy for optimizing the investment value is as follows:

- Find an operator $\mathcal{L}_\lambda$ on ϕ which satisfies

$$d[\phi(D(t))e^{-\lambda t}] = \mathcal{L}_\lambda\phi(D(s))e^{-\lambda t}dt + \sigma_D\phi'(D(t))D(t)e^{-\lambda t}dB(t). \tag{5.1}$$

- Find a unique bounded solution ϕ of $\mathcal{L}_r\phi = \min(x, S)$ with $|x\phi'(x)|$ bounded on $[0, \infty)$. Apply Theorem 5.1 to yield an expression for $E\left[\int_0^\tau \min(D(t), S_0)e^{-rt}dt\right]$. Since

$$0 \leq C \quad E\left[\int_0^\infty \min[D(s), S_1]e^{-rs}ds\right] \leq E\left[S_1 \int_0^\infty e^{-rs}ds\right] < \infty,$$

we also have an expression for $E\left[\int_\tau^\infty \min(D(t), S_1)e^{-rt}dt\right]$

$$E\left[\int_\tau^\infty \min(D(t), S_1)e^{-rt}dt\right] = C - E\left[\int_0^\tau \min(D(t), S_1)e^{-rt}dt\right], \tag{5.2}$$

and $0 \leq C \leq \frac{S_1}{r}$.

- Find a unique bounded solution ϕ of $\mathcal{L}_\lambda\phi \equiv 0$ with $|x\phi'(x)|$ bounded on $[0, y]$. Apply Theorem 5.2 to yield expressions for the discount factors $E[e^{-r\tau}]$ and $E[e^{-(r+\alpha)\tau}]$.

- Collect all the terms together in Theorem 5.3 to obtain an expression for investment value $F(D, y)$.

- Maximize $F(D, y)$ to obtain the optimal trigger y^* and investment value $F(D, y^*)$.

Theorem 5.1. *If ϕ is the unique bounded solution of*

$$\mathcal{L}_r\phi(x) = \min(x, S),$$

where $\mathcal{L}_\lambda\phi(x)$ satisfies (5.1) and $|x\phi'(x)|$ is bounded on $[0, \infty)$, then

$$E\left[\int_0^\tau \min(D(t), S)e^{-rt}dt\right] = \phi(y)E[e^{-r\tau}] - \phi(D).$$

Proof. Integrating from 0 to t

$$\phi(D(t))e^{-rt} - \phi(D) = \int_0^t e^{-rs}[\mathcal{L}_r\phi(D(s))]ds + \int_0^t \sigma_D\phi'(D(s))D(s)e^{-rs}dB(s).$$

Recall that $\mathcal{L}_r\phi(x) = \min[x, S]$ and put $t = \tau \wedge n$, then

$$\phi(D(\tau\wedge n))e^{-r(\tau\wedge n)} - \phi(D) = \int_0^{\tau\wedge n} \min[D(s), S]e^{-rs}ds + \int_0^{\tau\wedge n} \sigma_D\phi'(D(s))D(s)e^{-rs}dB(s).$$

Taking expectations

$$\begin{aligned} E\left[\phi(D(\tau\wedge n))e^{-r(\tau\wedge n)}\right] - \phi(D) &= E\left[\int_0^{\tau\wedge n} \min[D(s), S]e^{-rs}ds\right] \\ &\quad + E\left[\int_0^{\tau\wedge n} \sigma_D\phi'(D(s))D(s)e^{-rs}dB(s)\right]. \end{aligned}$$

Note that $M_t = \int_0^t \sigma_D D(s)\phi'(D(s))e^{-rs}dB(s)$ is a martingale as

$$\begin{aligned} E\left[|M_t|\right] &\leq E\left[|M_t|^2\right]^{\frac{1}{2}} \\ &= E\left[\int_0^t \sigma_D^2 E\left[D(s)^2(\phi'(D(s))^2)\right] e^{-2rs}ds\right]^{\frac{1}{2}}, \end{aligned}$$

and since $|x\phi'(x)|$ is bounded

$$E\left[\int_0^t \sigma_D^2 E\left[D(s)^2(\phi'(D(s))^2)\right] e^{-2rs}ds\right]^{\frac{1}{2}} \leq C\left[\int_0^t \sigma_D^2 e^{-2rs}ds\right]^{\frac{1}{2}} < \infty.$$

By the optional sampling theorem

$$E\left[\int_0^{\tau\wedge n} \sigma_D\phi'(D(s))D(s)e^{-rs}dB(s)\right] = 0.$$

When $\tau < \infty$,

$$\begin{aligned} \lim_{n\to\infty} \tau\wedge n &= \tau, \\ \lim_{n\to\infty} \phi(D(\tau\wedge n)) &= \phi(D(\tau)) = \phi(y), \end{aligned}$$

as ϕ is continuous, and $t \to D(t)$ can be assumed to be continuous along each sample path, as B is. When $\tau = \infty$, $\tau \wedge n = n$ and

$$|\phi(D(n))e^{-rn}| \to 0 \text{ as } n \to \infty,$$

since ϕ is bounded and $r > 0$. Also

$$|\phi(D(\tau\wedge n))e^{-r\tau}| < |\phi(D(\tau\wedge n))| \leq C < \infty,$$

and so by Lebesgue dominated convergence

$$\lim_{n\to\infty} E\left[\phi(D(\tau\wedge n))e^{-r(\tau\wedge n)}\right] = \phi(y)E\left[e^{-r\tau}\right].$$

Let $1(\cdot)$ denote the indicator function then

$$E\left[\int_0^{\tau\wedge n} \min[D(s),S]e^{-rs}ds\right] = E\left[\int_0^{\infty} 1(s\leq\tau\wedge n)\min[D(s),S]e^{-rs}ds\right].$$

Taking limits as $n\to\infty$

$$\lim_{n\to\infty} 1(s\leq\tau\wedge n)\min[D(s),S]e^{-rs} = 1(s\leq\tau)\min[D(s),S]e^{-rs},$$

by considering $\tau<\infty$ and $\tau=\infty$ as above.

$$|1(s\leq\tau\wedge n)\min[D(s),S]e^{-rs}| \leq Se^{-rs},$$

and $0\leq\int_0^\infty Se^{-rs}ds<\infty$. By Lebesgue dominated convergence

$$\lim_{n\to\infty} E\left[\int_0^{\tau\wedge n} \min[D(s),S]e^{-rs}ds\right] = E\left[\int_0^{\tau} \min[D(s),S]e^{-rs}ds\right].$$

Thus

$$E\left[\int_0^{\tau} \min[D(s),S]e^{-rs}ds\right] = \phi(y)E[e^{-r\tau}] - \phi(D).$$

□

Theorem 5.2. *If ϕ is the unique bounded solution of*

$$\mathcal{L}_\lambda\phi(x)\equiv 0,$$

where $\mathcal{L}_\lambda\phi(x)$ satisfies (5.1) and $|x\phi'(x)|$ is bounded on $[0,y)$, then

$$E\left[e^{-\lambda\tau}\right] = \frac{\phi(D)}{\phi(y)}. \tag{5.3}$$

Proof. For $\lambda>0$

$$\phi(D(t))e^{-\lambda t} = \phi(D) + \int_0^t [\mathcal{L}_\lambda\phi(D(s))]e^{-\lambda s}ds + \int_0^t \sigma_D\phi'(D(s))D(s)e^{-\lambda s}dB(s).$$

Since $\mathcal{L}_\lambda \phi(x) \equiv 0$

$$\phi(D(t))e^{-\lambda t} = \phi(D) + \int_0^t \sigma_D \phi'(D(s))D(s)e^{-\lambda s}dB(s).$$

Put $t = \tau \wedge n$, then

$$\begin{aligned}\phi(D(\tau \wedge n))e^{-\lambda(\tau\wedge n)} &= \phi(D) + \int_0^{\tau\wedge n} \sigma_D \phi'(D(s))D(s)e^{-\lambda s}dB(s) \\ &= \phi(D) + \int_0^{\tau\wedge n} \sigma_D \min[\phi'(D(s))D(s), C_1]e^{-\lambda s}dB(s).\end{aligned}$$

But

$$t \to M_t = \int_0^t \sigma_D \min[\phi'(D(s))D(s), C_1]e^{-\lambda s}dB(s),$$

is a martingale for each t:

$$\begin{aligned}E[|M_t|] &\leq \left[\int_0^t \sigma_D^2 C_1^2 ds\right]^{\frac{1}{2}} \\ &\leq t\sigma_D C_1 < \infty.\end{aligned}$$

So $E[\phi(D(\tau\wedge n))e^{-\lambda(\tau\wedge n)}] = \phi(D)$, then taking limits with the same argument above:

$$\phi(y)E\left[e^{-\lambda\tau}\right] = \phi(D).$$

□

Theorem 5.3.

$$F(D,y) = \beta\left\{\phi_0(y)E[e^{-r\tau}] - \phi_0(D) + C - \phi_1(y)E[e^{-r\tau}] + \phi_1(D)\right\} - IE[e^{-(r+\alpha)\tau}],$$

where ϕ_0 , ϕ_1 *are bounded* C^2 *solutions of*

$$\begin{aligned}\mathcal{L}_r\phi_0(x) &= \min(x, S_0), \\ \mathcal{L}_r\phi_1(x) &= \min(x, S_1),\end{aligned}$$

$C = E\left[\int_0^\infty \min(D(t), S_1)e^{-rt}dt\right]$ *and* $E[e^{-\lambda\tau}]$ *is given by (5.3).*

Proof. Applying Theorem 5.1 leads to

$$E\left[\int_0^{\tau} \min(D(t), S_0)e^{-rt}dt\right] = \phi_0(y)E[e^{-r\tau}] - \phi_0(D).$$

Applying Equation (5.2) and Theorem 5.1 leads to

$$E\left[\int_{\tau}^{\infty} \min(D(t), S_1)e^{-rt}dt\right] = C - \phi_1(y)E[e^{-r\tau}] + \phi_1(D),$$

where $C = E\left[\int_0^{\infty} \min(D(t), S_1)e^{-rt}dt\right]$. Collecting these results yields the expression for $F(D, y)$. □

5.3 A Geometric Brownian Motion Model

In this section we start with the simplest case where the demand process $D(t)$ is assumed to follow a geometric Brownian motion

$$dD(t) = \nu_D D(t)dt + \sigma_D D(t)dB(t), \tag{5.4}$$

where ν_D and σ_D are the drift and volatility terms. Sections 5.3.1-5.3.4 are used to derive the investment value $F(D, y)$. Section 5.3.5 explains how the optimal trigger is found, and Section 5.3.6 gives some numerical examples. From (2.4), we know that the stopping time τ is almost surely finite when $\nu_D - \frac{\sigma_D^2}{2} > 0$.

5.3.1 An expression for $\mathcal{L}_\lambda\phi(x)$

We seek an expression for $\mathcal{L}_\lambda\phi(x)$ which satisfies (5.1), where $\mathcal{L}_\lambda$ is an operator applied to the function $\phi(x)$. Lemma 5.4 provides an expression for the demand process defined in (5.4).

Lemma 5.4. *If the demand process $D(t)$ follows a geometric Brownian motion*

$$dD(t) = \nu_D D(t)dt + \sigma_D D(t)dB(t)$$

then $\mathcal{L}_\lambda\phi = \frac{1}{2}\sigma_D^2 x^2\phi''(x) + \nu_D x\phi'(x) - \lambda\phi(x)$ *satisfies (5.1).*

Proof. By Itô's lemma, we have for any C^2 function ϕ

$$\begin{aligned} d[\phi(D(t))e^{-\lambda t}] &= -\lambda\phi(D(t))e^{-\lambda t}dt + e^{-\lambda t}\phi'(D(t))dD(t) + e^{-\lambda t}\phi''(D(t))\frac{1}{2}dD(t)^2 \\ &= -\lambda\phi(D(t))e^{-\lambda t}dt + e^{-\lambda t}\phi'(D(t))[\nu_D D(t)dt + \sigma_D D(t)dB(t)] \\ &\quad +e^{-\lambda t}\phi''(D(t))\frac{\sigma_D^2}{2}D(t)^2dt \\ &= e^{-\lambda t}\left\{-\lambda\phi(D(t)) + \nu_D D(t)\phi(D(t))' + \frac{\sigma_D^2}{2}D(t)^2\phi''(D(t))\right\}dt \\ &\quad +\sigma_D e^{-\lambda t}\phi'(D(t))D(t)dB(t). \end{aligned}$$

Let $\mathcal{L}_\lambda\phi(x) = -\lambda\phi(x) + \nu_D x\phi'(x) + \frac{1}{2}\sigma_D^2 x^2\phi''(x)$,

$$d[\phi(D(t))e^{-\lambda t}] = \mathcal{L}_\lambda\phi(D(s))e^{-\lambda t}dt + \sigma_D\phi'(D(t))D(t)e^{-\lambda t}dB(t).$$

□

5.3.2 A unique bounded solution for $\mathcal{L}_r\phi(x) = \min(x, S)$

Lemma 5.5 provides a unique bounded solution $\phi(x)$ which satisfies $\mathcal{L}_r\phi(x) = \min(x, S)$ when $\nu_D \neq r$. Lemma 5.6 provides an alternative solution for the special case $\nu_D = r$. Henceforth, we shall let $\phi_0(x)$ and $\phi_1(x)$ denote the solutions for $S = S_0$ and $S = S_1$ respectively. Figure 5.1 shows the ϕ functions for Example 5.1a (see Table 5.1).

Lemma 5.5. *If $\nu_D \neq r$, the unique bounded solution of $\mathcal{L}_r\phi(x) = \min(x, S)$ is:*

$$\phi(x) = \begin{cases} \frac{x}{\nu_D - r} + Ax^{k_1}, & \text{if } x \leq S; \\ \frac{-S}{r} + Bx^{k_2}, & \text{if } x > S, \end{cases} \quad (5.5)$$

$$A = \frac{r - \nu_D k_2}{r(\nu_D - r)(k_2 - k_1)S^{k_1 - 1}}, \quad (5.6)$$

$$B = \frac{r - \nu_D k_1}{r(\nu_D - r)(k_2 - k_1)S^{k_2 - 1}}, \quad (5.7)$$

and k_1 and k_2 are the positive and negative roots of

$$\frac{1}{2}\sigma_D^2 k(k-1) + \nu_D k - r = 0.$$

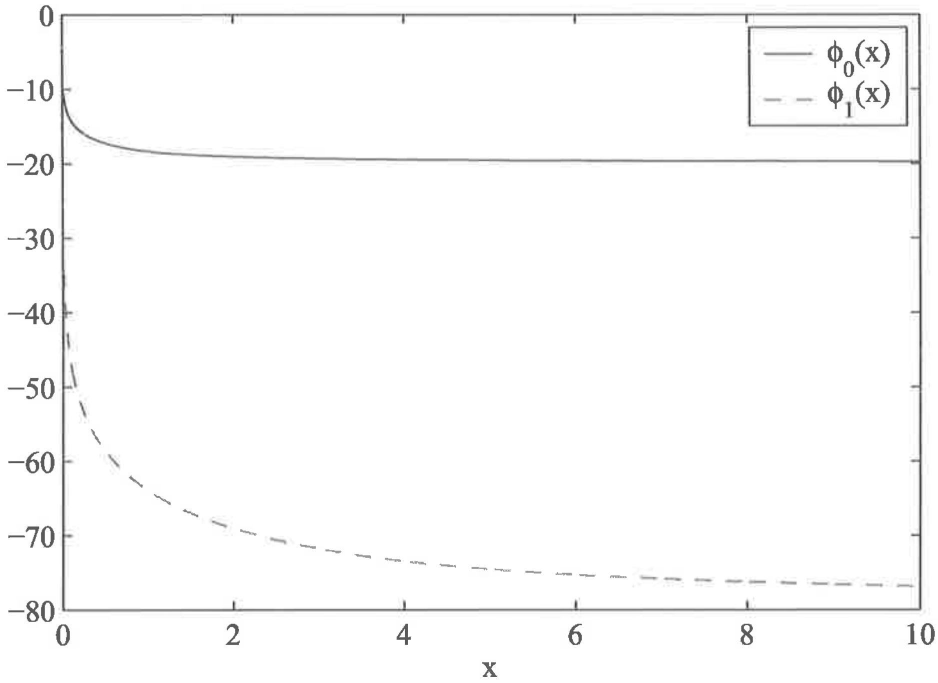


Figure 5.1: The ϕ functions for Example 5.1a

Proof. We seek a unique, bounded solution of

$$\frac{1}{2}\sigma_D^2 x^2 \phi''(x) + \nu_D x \phi'(x) - r\phi = \begin{cases} x, & \text{if } x \leq S; \\ S, & \text{if } x > S, \end{cases} \tag{5.8}$$

with $|x\phi'(x)|$ bounded. For $x \leq S$,

$$\frac{1}{2}\sigma_D^2 x^2 \phi''(x) + \nu_D x \phi'(x) - r\phi = x. \tag{5.9}$$

The homogeneous solution is

$$\phi(x) = A_1 x^{k_1} + A_2 x^{k_2},$$

where k_1 and k_2 be the positive and negative roots of the characteristic equation

$$\frac{1}{2}\sigma_D^2 k(k-1) + \nu_D k - r = 0. \tag{5.10}$$

The particular solution is

$$\phi(x) = \frac{x}{\nu_D - r}.$$

In order that ϕ be bounded near 0, $A_2 = 0$ ($\phi(0) = 0$ from (5.9)), and so

$$\phi(x) = \frac{x}{\nu_D - r} + Ax^{k_1},$$

for $x \leq S$, which establishes the first part of (5.5).

For $x > S$,

$$\frac{1}{2}\sigma_D^2 x^2 \phi''(x) + \nu_D x \phi'(x) - r\phi = S.$$

The homogeneous solution is

$$\phi(x) = B_1 x^{k_1} + B_2 x^{k_2}.$$

The particular solution is

$$\phi(x) = \frac{-S}{r}.$$

For ϕ bounded, we require $B_1 = 0$, and so

$$\phi(x) = \frac{-S}{r} + Bx^{k_2},$$

for $x \geq S$, which establishes the second part of (5.5).

For ϕ smooth, $\phi(S^-) = \phi(S^+)$ and $\phi'(S^-) = \phi'(S^+)$,

$$\frac{S}{\nu_D - r} + AS^{k_1} = \frac{-S}{r} + BS^{k_2}, \tag{5.11}$$

$$\frac{1}{\nu_D - r} + Ak_1 S^{k_1 - 1} = Bk_2 S^{k_2 - 1}. \tag{5.12}$$

Re-arranging (5.11) leads to

$$\begin{aligned} A &= \frac{1}{S^{k_1}}\left[BS^{k_2} - \frac{S}{\nu_D - r} - \frac{S}{r}\right] \\ &= BS^{k_2 - k_1} - \frac{\nu_D}{r(\nu_D - r)S^{k_1 - 1}}. \end{aligned} \tag{5.13}$$

Substituting this expression into (5.12) yields an expression for B

$$\begin{aligned} Bk_2 S^{k_2 - 1} &= \frac{1}{\nu_D - r} + \left[BS^{k_2 - k_1} - \frac{\nu_D}{r(\nu_D - r)S^{k_1 - 1}}\right] k_1 S^{k_1 - 1}, \\ B(k_2 - k_1)S^{k_2 - 1} &= \frac{1}{\nu_D - r} - \frac{\nu_D k_1}{r(\nu_D - r)} \\ B &= \frac{r - \nu_D k_1}{r(\nu_D - r)(k_2 - k_1)S^{k_2 - 1}}. \end{aligned}$$

Finally, substituting this expression into (5.13) yields an expression for A

$$\begin{aligned} A &= \frac{r-\nu_D k_1}{r(\nu_D-r)(k_2-k_1)S^{k_2-1}} S^{k_2-k_1} - \frac{\nu_D}{r(\nu_D-r)S^{k_1-1}} \\ &= \frac{r-\nu_D k_1}{r(\nu_D-r)(k_2-k_1)S^{k_1-1}} - \frac{\nu_D}{r(\nu_D-r)S^{k_1-1}} \\ &= \frac{r-\nu_D k_2}{r(\nu_D-r)(k_2-k_1)S^{k_1-1}}. \end{aligned}$$

We note that $|x\phi'(x)| \leq C$ for all $x \geq 0$. For $x \in [0, S]$,

$$\begin{aligned} |x\phi'(x)| &= \left| \frac{x}{\nu_D - r} + k_1 A x^{k_1} \right| \\ &\leq \frac{S}{|\nu_D - r|} + k_1 |A| S^{k_1} < \infty. \end{aligned}$$

For $x \in [S, \infty)$,

$$\begin{aligned} |x\phi'(x)| &= \left| k_2 B x^{k_2} \right| \\ &= |k_2||B|x^{k_2} \\ &\leq |k_2||B|S^{k_2} \text{ (as } k_2 < 0 \text{)} \\ &< \infty. \end{aligned}$$

So

$$|x\phi'(x)| \leq \max\left[\frac{S}{|\nu_D - r|} + k_1|A|S^{k_1}, |k_2||B|S^{k_2}\right] < \infty.$$

We also note that ϕ is twice differentiable. We only need to show that the second derivatives match at S,

$$\phi''(S-) = \phi''(S+).$$

Re-arranging the PDE

$$\phi''(x) = \frac{\min(S, x) + r\phi(x) - \nu_D x\phi'(x)}{\frac{1}{2}x^2\sigma^2}, \quad x \neq S.$$

Since

$$\begin{aligned} \phi(S+) &= \phi(S-), \\ \phi'(S+) &= \phi'(S-), \end{aligned}$$

we must have $\phi''(S+) = \phi''(S-)$.

Finally, we note that ϕ is unique. Suppose that ϕ_1 and ϕ_2 are two solutions. Let

$$\overline{\phi} = \phi_1 - \phi_2,$$

then

$$\mathcal{L}_r\overline{\phi} = 0.$$

The general solution is

$$\overline{\phi}(x) = Ax^{k_1} + Bx^{k_2}.$$

Since $\overline{\phi}$ bounded, we must have $B = 0$. Since $\overline{\phi}(0) = 0$, we must have $A = 0$. Therefore $\overline{\phi} = 0$. □

Lemma 5.6. *If $\nu_D = r$, the unique bounded solution of $\mathcal{L}_r\phi(x) = \min(x, S)$ is:*

$$\phi(x) = \begin{cases} \frac{x\ln(x)}{\frac{1}{2}\sigma_D^2+r} + Ax, & \text{if } x \leq S; \\ \frac{-S}{r} + Bx^{\frac{-2r}{\sigma_D^2}}, & \text{if } x > S, \end{cases} \tag{5.14}$$

$$A = \frac{(\ln(S)+1)(1+\frac{2r}{\sigma_D^2})+1}{-(\frac{1}{2}\sigma_D^2+r)(\frac{2r}{\sigma_D^2}+1)}, \tag{5.15}$$

$$B = \frac{\frac{1}{2}\sigma_D^2 S^{\frac{2r}{\sigma_D^2}+1}}{r(\frac{1}{2}\sigma_D^2+r)(\frac{2r}{\sigma_D^2}+1)}. \tag{5.16}$$

Proof. We seek a unique, bounded solution of

$$\frac{1}{2}\sigma_D^2 x^2\phi''(x) + rx\phi'(x) - r\phi(x) = \begin{cases} x, & \text{if } x \leq S; \\ S, & \text{if } x > S, \end{cases} \tag{5.17}$$

with $|x\phi'(x)|$ bounded. Note that the only difference between (5.8) and (5.17) is that ν_D has been replaced by r. For $x \leq S$,

$$\frac{1}{2}\sigma_D^2 x^2\phi''(x) + rx\phi'(x) - r\phi(x) = x. \tag{5.18}$$

The homogeneous solution is

$$\phi(x) = A_1x^{k_1} + A_2x^{k_2},$$

where k_1 and k_2 are the positive and negative roots of the characteristic equation

$$\left(\frac{1}{2}\sigma_D^2 k + r\right)(k-1) = 0. \tag{5.19}$$

In fact, $k_1 = 1$ and $k_2 = \frac{-2r}{\sigma_D^2}$. Particular solutions of the form $\phi(x) = cx$ do not apply here, so we use $\phi(x) = cx\ln(x)$:

$$\begin{aligned}\frac{1}{2}\sigma_D^2 x^2 \phi''(x) + rx\phi'(x) - r\phi(x) &= \frac{1}{2}\sigma_D^2 x^2\left(\frac{c}{x}\right) + rx(c\ln(x)+c) - r(c\ln(x)) \\ &= \left(\frac{1}{2}\sigma_D^2 + r\right)cx.\end{aligned}$$

Hence the particular solution is

$$\phi(x) = \frac{x\ln(x)}{\frac{1}{2}\sigma_D^2 + r}.$$

In order that ϕ be bounded near 0, $A_2 = 0$ ($\phi(0) = 0$ from (5.18)), and so

$$\phi(x) = \frac{x\ln(x)}{\frac{1}{2}\sigma_D^2 + r} + Ax^{k_1},$$

for $x \le S$, which establishes the first part of (5.14). For $x > S$,

$$\frac{1}{2}\sigma_D^2 x^2 \phi''(x) + \nu_D x\phi'(x) - r\phi = S.$$

The homogeneous solution is

$$\phi(x) = B_1 x^{k_1} + B_2 x^{k_2}.$$

The particular solution is

$$\phi(x) = \frac{-S}{r}.$$

For ϕ bounded, we require $B_1 = 0$, and so

$$\phi(x) = \frac{-S}{r} + Bx^{k_2},$$

for $x \ge S$, which establishes the second part of (5.14).

For ϕ smooth, $\phi(S^-) = \phi(S^+)$ and $\phi'(S^-) = \phi'(S^+)$,

$$\frac{S\ln(S)}{\frac{1}{2}\sigma_D^2 + r} + AS^{k_1} = \frac{-S}{r} + BS^{k_2}, \tag{5.20}$$

$$\frac{\ln(S)+1}{\frac{1}{2}\sigma_D^2 + r} + Ak_1 S^{k_1-1} = Bk_2 S^{k_2-1}. \tag{5.21}$$

Re-arranging (5.20) leads to

$$\begin{aligned} A &= \frac{1}{S^{k_1}}\left[BS^{k_2} - \frac{S\ln(S)}{\frac{1}{2}\sigma_D^2 + r} - \frac{S}{r}\right] \\ &= BS^{k_2-k_1} - \frac{r\ln(S) + \frac{1}{2}\sigma_D^2 + r}{r(\frac{1}{2}\sigma_D^2 + r)S^{k_1-1}}. \end{aligned} \tag{5.22}$$

Substituting this expression into (5.21) yields an expression for B

$$\begin{aligned} Bk_2S^{k_2-1} &= \frac{\ln(S)+1}{\frac{1}{2}\sigma_D^2 + r} + \left[BS^{k_2-k_1} - \frac{r\ln(S) + \frac{1}{2}\sigma_D^2 + r}{r(\frac{1}{2}\sigma_D^2 + r)S^{k_1-1}}\right] k_1 S^{k_1-1}, \\ B(k_2-k_1)S^{k_2-1} &= \frac{\ln(S)+1}{\frac{1}{2}\sigma_D^2 + r} - \frac{k_1(r\ln(S) + \frac{1}{2}\sigma_D^2 + r)}{r(\frac{1}{2}\sigma_D^2 + r)} \\ B &= \frac{r(\ln(S)+1)(1-k_1) - \frac{1}{2}\sigma_D^2 k_1}{r(\frac{1}{2}\sigma_D^2 + r)(k_2-k_1)S^{k_2-1}}. \end{aligned}$$

Finally, substituting this expression into (5.22) yields an expression for A

$$\begin{aligned} A &= \frac{r(\ln(S)+1)(1-k_1) - \frac{1}{2}\sigma_D^2 k_1}{r(\frac{1}{2}\sigma_D^2 + r)(k_2-k_1)S^{k_2-1}} S^{k_2-k_1} - \frac{r\ln(S) + \frac{1}{2}\sigma_D^2 + r}{r(\frac{1}{2}\sigma_D^2 + r)S^{k_1-1}} \\ &= \frac{r(\ln(S)+1)(1-k_1) - \frac{1}{2}\sigma_D^2 k_1}{r(\frac{1}{2}\sigma_D^2 + r)(k_2-k_1)S^{k_1-1}} - \frac{r(\ln(S)+1) + \frac{1}{2}\sigma_D^2}{r(\frac{1}{2}\sigma_D^2 + r)S^{k_1-1}} \\ &= \frac{r(\ln(S)+1)(1-k_2) - \frac{1}{2}\sigma_D^2 k_2}{r(\frac{1}{2}\sigma_D^2 + r)(k_2-k_1)S^{k_1-1}}. \end{aligned}$$

Substituting $k_1 = 1$ and $k_2 = \frac{-2r}{\sigma_D^2}$ leads to

$$\begin{aligned} A &= \frac{(\ln(S)+1)(1+\frac{2r}{\sigma_D^2}) + 1}{-(\frac{1}{2}\sigma_D^2 + r)(\frac{2r}{\sigma_D^2} + 1)}, \\ B &= \frac{\frac{1}{2}\sigma_D^2 S^{\frac{2r}{\sigma_D^2}+1}}{r(\frac{1}{2}\sigma_D^2 + r)(\frac{2r}{\sigma_D^2} + 1)}. \end{aligned}$$

We note that $|x\phi'(x)| \leq C$ for all $x \geq 0$. For $x \in [0, S]$,

$$\begin{aligned} |x\phi'(x)| &= \left|\frac{x(\ln(x)+1)}{\frac{1}{2}\sigma_D^2 + r} + k_1 A x^{k_1}\right| \\ &\leq \frac{S(\ln(S)+1)}{\frac{1}{2}\sigma_D^2 + r} + k_1|A|S^{k_1} < \infty. \end{aligned}$$

For $x \in [S, \infty)$,

$$\begin{aligned} |x\phi'(x)| &= \left|k_2 B x^{k_2}\right| \\ &= |k_2||B|x^{k_2} \\ &\leq |k_2||B|S^{k_2} \text{ (as } k_2 < 0 \text{)} \\ &< \infty. \end{aligned}$$

So

$$|x\phi'(x)| \leq \max\left[\frac{S(\ln(S)+1)}{\frac{1}{2}\sigma_D^2 + r} + k_1|A|S^{k_1}, |k_2||B|S^{k_2}\right] < \infty.$$

We also note that ϕ is unique and twice differentiable. This can be shown using similar arguments to those given in the proof of Lemma 5.5. □

5.3.3 The discount factors $E\left[e^{-r\tau}\right]$ and $E\left[e^{-(r+\alpha)\tau}\right]$

Lemma 5.7 provides expressions for the discount factors $E[e^{-r\tau}]$ and $E[e^{-(r+\alpha)\tau}]$

$$\begin{aligned} E[e^{-r\tau}] &= \left(\frac{D}{y}\right)^{k_1}, \\ E[e^{-(r+\alpha)\tau}] &= \left(\frac{D}{y}\right)^{k_3}, \end{aligned}$$

where

$$\begin{aligned} k_1 &= \frac{-\left[\nu_D - \frac{\sigma_D^2}{2}\right] + \sqrt{\left(\nu_D - \frac{\sigma_D^2}{2}\right)^2 + 2r\sigma_D^2}}{\sigma_D^2}, \\ k_3 &= \frac{-\left[\nu_D - \frac{\sigma_D^2}{2}\right] + \sqrt{\left(\nu_D - \frac{\sigma_D^2}{2}\right)^2 + 2(r+\alpha)\sigma_D^2}}{\sigma_D^2}. \end{aligned}$$

Figure 5.2 shows the discount factors for Example 5.1a (see Table 5.1).

We note that the discount factor in Lemma 5.7 is equivalent to the discount factor $E[e^{-\lambda\tau}]$, which we derived for a GBM value process $V(t)$ in Section 3.2,

$$E[e^{-\lambda\tau}] = \left(\frac{V}{V*}\right)^{\frac{1}{\sigma}\left(-\mu+\sqrt{\mu^2+2\lambda}\right)},$$

where $\mu = \frac{\nu}{\sigma} - \frac{\sigma}{2}$.

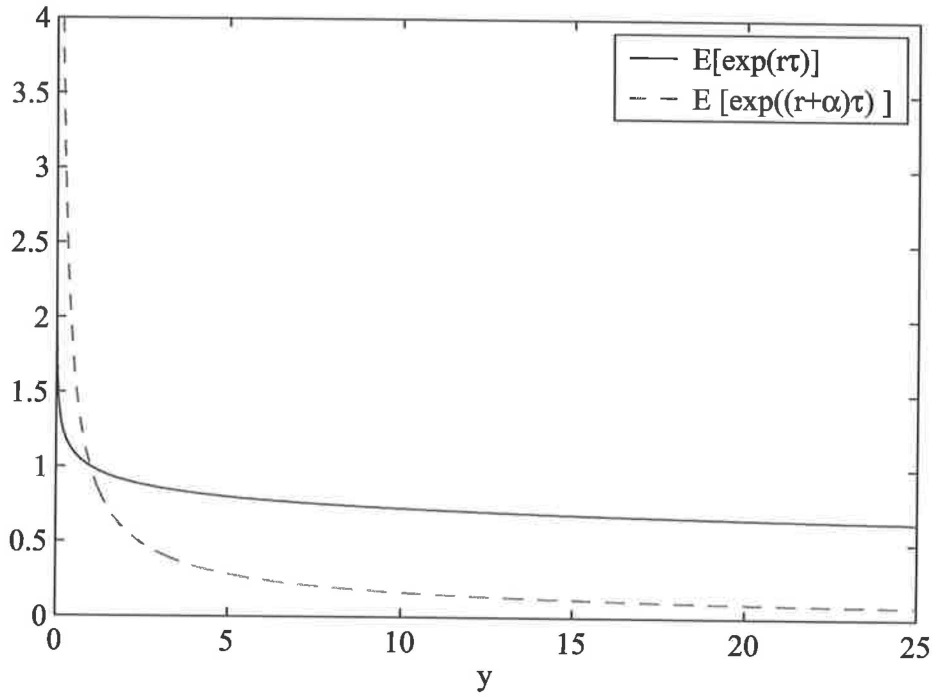


Figure 5.2: Discount factors for Example 5.1a

Lemma 5.7.

$$E[e^{-\lambda\tau}] = \left(\frac{D}{y}\right)^{\frac{1}{\sigma_D}\left(-\mu+\sqrt{\mu^2+2\lambda}\right)},$$

where $\mu = \frac{\nu_D}{\sigma_D} - \frac{\sigma_D}{2}$.

Proof. We seek a unique, bounded solution of

$$\frac{1}{2}\sigma_D^2 x^2 \phi''(x) + \nu_D x \phi'(x) - \lambda\phi(x) = 0,$$

with $|x\phi'(x)|$ bounded. The solution is

$$\phi(x) = Ax^k,$$

where $A > 0$ and k is the positive root of

$$\frac{1}{2}\sigma_D^2 k(k-1) + \nu_D k - \lambda = 0.$$

Hence

$$\begin{aligned} k &= \frac{-\left[\nu_D - \frac{\sigma_D^2}{2}\right] + \sqrt{\left(\nu_D - \frac{\sigma_D^2}{2}\right)^2 + 2\lambda\sigma_D^2}}{\sigma_D^2} \\ &= \frac{-\mu + \sqrt{\mu^2 + 2\lambda}}{\sigma_D}, \end{aligned}$$

where $\mu = \frac{\nu_D}{\sigma_D} - \frac{\sigma_D}{2}$. For x on $[0, y]$,

$$|x\phi'(x)| \leq k|A|y^k = C_1 < \infty.$$

Applying Theorem 5.2,

$$\begin{aligned} E\left[e^{-\lambda\tau}\right] &= \frac{\phi(D)}{\phi(y)} \\ &= \frac{AD^k}{Ay^k} \\ &= \left(\frac{D}{y}\right)^k. \end{aligned}$$

□

5.3.4 An expression for $E\left[\int_0^\infty \min(D(s), S)e^{-rs}ds\right]$

Theorem 5.8 provides an expression for $E_t\left[\int_t^\infty \min(D(s), S)e^{-r(s-t)}ds\right]$. Setting $t = 0$ provides an expression for

$$C = E\left[\int_0^\infty \min(D(s), S)e^{-rs}ds\right].$$

Since there is no closed form solution, numerical integration techniques (e.g. Simpson's rule) can be employed.

Theorem 5.8.

$$E_t\left[\int_t^\infty \min(D(s), S)e^{-r(s-t)}ds\right] = \int_t^\infty \left\{D(t)e^{(\nu_D - r)(s-t)}N(-d_1(s)) + Se^{-r(s-t)}N(d_2(s))\right\} ds$$

where

$$\begin{aligned} d_1(s) &= \frac{\log(D(t)/S) + (\nu_D + \sigma^2/2)(s-t)}{\sigma\sqrt{s-t}}, \\ d_2(s) &= d_1(s) - \sigma\sqrt{s-t}, \\ N(x) &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^x \exp\left(-\frac{z^2}{2}\right) dz. \end{aligned}$$

Proof. We seek

$$V(t) = E_t\left[\int_t^\infty \min(D(s), S)\exp(-r(s-t))ds\right].$$

We can take the expectation inside the integral and note that

$$\min(D(s), S) = D(s) - (D(s) - S)^+,$$

so

$$\begin{aligned} dV(s) &= \{E_t[D(s)e^{-r(s-t)}] - E_t[(D(s)-S)^+e^{-r(s-t)}]\}ds \\ &= \{D(t)e^{(\nu_D - r)(s-t)} - E_t[(D(s)-S)^+e^{-r(s-t)}]\}ds. \end{aligned}$$

We note that $E_t[(D(s)-S)^+e^{-r(s-t)}]$ is like a call option and so

$$E_t[(D(s)-S)^+e^{-r(s-t)}] = D(t)e^{(\nu_D - r)(s-t)}N(d_1(s)) - Se^{-r(s-t)}N(d_2(s)),$$

where

$$d_1(s) = \frac{\log(D(t)/S) + (\nu_D + \sigma^2/2)(s-t)}{\sigma\sqrt{s-t}},$$

and $d_2(s) = d1(s) - \sigma\sqrt{s-t}$. Combining these two results we have

$$\begin{aligned} dV(s) &= \left\{D(t)e^{(\nu_D - r)(s-t)} - De^{(\nu_D - r)(s-t)}N(d_1(s)) + Se^{-r(s-t)}N(d_2(s))\right\}ds \\ &= \left\{D(t)e^{(\nu_D - r)(s-t)}N(-d_1(s)) + Se^{-r(s-t)}N(d_2(s))\right\}ds. \end{aligned}$$

Thus

$$V(t) = \int_t^\infty \left\{D(t)e^{(\nu_D - r)(s-t)}N(-d_1(s)) + Se^{-r(s-t)}N(d_2(s))\right\}ds.$$

□

Lemma 5.9.

$$E\left[\int_0^\infty \min(D(s), S)e^{-rs}ds\right] = \int_0^\infty \left\{De^{(\nu_D - r)s}N(-d_1(s)) + Se^{-rs}N(d_2(s))\right\}ds,$$

where

$$\begin{aligned} d_1(s) &= \frac{\log(D/S) + (\nu_D + \sigma^2/2)s}{\sigma\sqrt{s}}, \\ d_2(s) &= d_1(s) - \sigma\sqrt{s}, \\ N(x) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{z^2}{2}\right) dz. \end{aligned}$$

Proof. Set $t := 0$ in Theorem 5.8. □

5.3.5 The Optimal Trigger

In this section we shall find the optimal trigger y^* for $F(D, y)$. The investment function $F(D, y)$ has a distinct form on the following three regions: $[0, S_0]$, $[S_0, S_1]$ and $[S_1, \infty)$. A local maximum may be found by applying the bisection method to $\frac{\partial F(D,y)}{\partial y} = 0$, but we must first establish uniqueness and existence. We would also like to determine the region in which the optimal trigger y^* will reside. Lemma 5.10 shows that y^* must be greater than the initial transmission capacity (S_0) so y^* may be in the second region (S_0, S_1) or the third region (S_1, ∞). Theorem 5.16 shows that the optimal trigger exists and is unique, and Algorithm 5.17 provides a method for finding y^*.

Lemma 5.10. *The optimal trigger y^* is not in $[0, S_0]$.*

Proof. We can establish this result by first showing that

$$F(D, y) = C_1 - I\left(\frac{D}{y}\right)^{k_3} \quad \forall y \in [0, S_0],$$

for some constant C_1. This implies that

$$\frac{\partial F(D, y)}{\partial y} = k_3 I D^{k_3} \left(\frac{1}{y}\right)^{k_3+1} > 0, \ \forall y \in [0, S_0],$$

and so the optimal trigger is not in $[0, S_0]$.

First, consider the case $\nu_D \neq r$,

$$\begin{aligned}
F(D,y) &= \beta\left(\frac{y}{\nu_D - r} + A_0 y^{k_1}\right)\left(\frac{D}{y}\right)^{k_1} - \beta\phi_0(D) + \beta C \\
&\quad -\beta\left(\frac{y}{\nu_D - r} + A_1 y^{k_1}\right)\left(\frac{D}{y}\right)^{k_1} + \beta\phi_1(D) - I\left(\frac{D}{y}\right)^{k_3} \\
&= \beta(A_0 D^{k_1} - \phi_0(D) + C - A_1 D^{k_1} + \phi_1(D)) - I\left(\frac{D}{y}\right)^{k_3} \\
&= C_1 - I\left(\frac{D}{y}\right)^{k_3},
\end{aligned}$$

where $C_1 = \beta(A_0 D^{k_1} - \phi_0(D) + C - A_1 D^{k_1} + \phi_1(D))$ and

$$\begin{aligned}
k_1 &= \frac{-\left[\nu_D - \frac{\sigma_D^2}{2}\right] + \sqrt{\left(\nu_D - \frac{\sigma_D^2}{2}\right)^2 + 2r\sigma_D^2}}{\sigma_D^2}, \\
k_2 &= \frac{-\left[\nu_D - \frac{\sigma_D^2}{2}\right] - \sqrt{\left(\nu_D - \frac{\sigma_D^2}{2}\right)^2 + 2r\sigma_D^2}}{\sigma_D^2}, \\
k_3 &= \frac{-\left[\nu_D - \frac{\sigma_D^2}{2}\right] + \sqrt{\left(\nu_D - \frac{\sigma_D^2}{2}\right)^2 + 2(r+\alpha)\sigma_D^2}}{\sigma_D^2}.
\end{aligned}$$

Now, consider the case $\nu_D = r$,

$$\begin{aligned}
F(D,y) &= \beta\left(\frac{y\ln(y)}{\frac{\sigma_D^2}{2} + r} + A_0 y\right)\left(\frac{D}{y}\right) - \beta\phi_0(D) + \beta C \\
&\quad -\beta\left(\frac{y\ln(y)}{\frac{\sigma_D^2}{2} + r} + A_1 y\right)\left(\frac{D}{y}\right) + \beta\phi_1(D) - I\left(\frac{D}{y}\right)^{k_3} \\
&= \beta(A_0 D^{k_1} - \phi_0(D) + C - A_1 D^{k_1} + \phi_1(D)) - I\left(\frac{D}{y}\right)^{k_3} \\
&= C_1 - I\left(\frac{D}{y}\right)^{k_3},
\end{aligned}$$

where $C_1 = \beta(A_0 D - \phi_0(D) + C - A_1 D + \phi_1(D))$ and

$$\begin{aligned}
k_1 &= 1, \\
k_2 &= \frac{-2r}{\sigma_D^2}, \\
k_3 &= \frac{-\left[r - \frac{\sigma_D^2}{2}\right] + \sqrt{\left(r + \frac{\sigma_D^2}{2}\right)^2 + 2\alpha\sigma_D^2}}{\sigma_D^2}.
\end{aligned}$$

□

Lemma 5.11. *For $y \in [S_0, S_1]$, we can write*

$$F'(D,y) = \frac{\beta}{y}\left(\frac{D}{y}\right)^{k_1} f_-(y),$$

where

$$f_-(y) = \begin{cases} \frac{k_1 S_0}{r} + \frac{(k_1-1)y}{\nu_D - r} + (k_2 - k_1)B_0 y^{k_2} + \frac{k_3 I}{\beta}\left(\frac{D}{y}\right)^{k_3-k_1}, & \nu_D \neq r; \\ \frac{S_0}{r} - \frac{2y}{\sigma_D^2+2r} + (\frac{-2r}{\sigma_D^2} - 1)B_0 y^{\frac{-2r}{\sigma_D^2}} + \frac{k_3 I}{\beta}\left(\frac{D}{y}\right)^{k_3-1}, & \nu_D = r. \end{cases}$$

Proof. First, consider the case $\nu_D \neq r$,

$$\begin{aligned} F(D,y) &= \beta\left(\frac{-S_0}{r} + B_0 y^{k_2}\right)\left(\frac{D}{y}\right)^{k_1} - \beta\phi_0(D) + \beta C \\ &\quad -\beta\left(\frac{y}{\nu_D - r} + A_1 y^{k_1}\right)\left(\frac{D}{y}\right)^{k_1} + \beta\phi_1(D) - I\left(\frac{D}{y}\right)^{k_3} \\ &= \beta(-\phi_0(D) + C - A_1 D^{k_1} + \phi_1(D)) \\ &\quad +\beta D^{k_1}\left(\frac{-S_0}{r}y^{-k_1} - \frac{1}{\nu_D - r}y^{-(k_1-1)} + B_0 y^{k_2-k_1}\right) \\ &\quad -I\left(\frac{D}{y}\right)^{k_3}. \end{aligned}$$

Thus

$$\begin{aligned} \frac{\partial F(D,y)}{\partial y} &= \frac{\beta}{y}\left(\frac{D}{y}\right)^{k_1}\left(\frac{k_1 S_0}{r} + \frac{(k_1-1)y}{\nu_D - r} + (k_2-k_1)B_0 y^{k_2-1}\right) \\ &\quad +k_3 I D^{k_3}\left(\frac{1}{y}\right)^{k_3+1} \\ &= \frac{\beta}{y}\left(\frac{D}{y}\right)^{k_1} f_-(y). \end{aligned}$$

Now consider the case $\nu_D = r$,

$$\begin{aligned} F(D,y) &= \beta\left(\frac{-S_0}{r} + B_0 y^{\frac{-2r}{\sigma_D^2}}\right)\frac{D}{y} - \beta\phi_0(D) + \beta C \\ &\quad -\beta\left(\frac{2y\ln(y)}{\sigma_D^2 + 2r} + A_1 y\right)\frac{D}{y} + \beta\phi_1(D) - I\left(\frac{D}{y}\right)^{k_3} \\ &= \beta(-\phi_0(D) + C - A_1 D + \phi_1(D)) \\ &\quad +\beta D^{k_1}\left(\frac{-S_0}{ry} - \frac{2\ln(y)}{\sigma_D^2 + 2r} + B_0 y^{\frac{-2r}{\sigma_D^2}-1}\right) \\ &\quad -I\left(\frac{D}{y}\right)^{k_3}. \end{aligned}$$

Thus

$$\begin{aligned}\frac{\partial F(D,y)}{\partial y} &= \frac{\beta}{y}\frac{D}{y}\left(\frac{S_0}{r} - \frac{2y}{\sigma_D^2 + 2r} + \left(\frac{-2r}{\sigma_D^2} - 1\right) B_0 y^{\frac{-2r}{\sigma_D^2} - 1}\right) \\ &\quad + k_3 I D^{k_3}\left(\frac{1}{y}\right)^{k_3+1} \\ &= \frac{\beta}{y}\left(\frac{D}{y}\right)^{k_1} f_-(y).\end{aligned}$$

□

Lemma 5.12. *If $f_-(y) = 0$ then $f_-'(y) < 0$.*

Proof. We need to establish three intermediate results:

$$r(k_1 + k_2 - 1) - \nu_D k_1 k_2 = 0, \tag{5.23}$$

$$\frac{(k_1 - 1)}{\nu_D - r} < 0, \tag{5.24}$$

$$\frac{r - \nu_D k_1}{r(\nu_D - r)} < 0. \tag{5.25}$$

First, consider (5.23)

$$\begin{aligned}r(k_1 + k_2 - 1) - \nu_D k_1 k_2 &= r\left(\frac{-2[\nu_D - \frac{\sigma_D^2}{2}]}{\sigma_D^2} - 1\right) - \nu_D \frac{[\nu_D - \frac{\sigma_D^2}{2}]^2 - \{(\nu_D - \frac{\sigma_D^2}{2})^2 + 2r\sigma_D^2\}}{\sigma_D^4} \\ &= \frac{r(-2\nu_D + \sigma_D^2 - \sigma_D^2) + 2\nu_D r}{\sigma_D^2} = 0.\end{aligned}$$

Second, consider (5.24). If $\nu_D > r$, we have $k_1 < 1$. Thus $k_1 - 1 < 0$ and $\nu_D - r > 0$, and so

$$\frac{(k_1 - 1)}{\nu_D - r} < 0.$$

Suppose instead that $\nu_D < r$, then $k_1 > 1$. Thus $k_1 - 1 > 0$ and $\nu_D - r < 0$, and so

$$\frac{(k_1 - 1)}{\nu_D - r} < 0.$$

Third, consider (5.25). Using (5.23) we know that

$$(r - \nu_D k_1)k_2 = -r(k_1 - 1).$$

Thus

$$\begin{aligned}\frac{r-\nu_D k_1}{r(\nu_D - r)} &= \frac{-r(k_1-1)}{rk_2(\nu_D - r)}\\ &= \frac{-1}{k_2}\frac{k_1-1}{\nu_D - r}.\end{aligned}$$

Since $\frac{-1}{k_2} > 0$ and $\frac{k_1-1}{\nu_D-r} < 0$, this expression must be negative. Since B is a product of two negative terms $\left(\frac{r-\nu_D k_1}{(\nu_D-r)} \text{ and } \frac{1}{k_2-k_1}\right)$ and some positive terms $\left(\frac{1}{r} \text{ and } S^{1-k_2}\right)$, this result also establishes that B_0 and B_1 are positive.

Now let's return to our original problem. We want to show that $f_-(y) = 0$ implies $f'_-(y) < 0$. In each case, we shall find an expression for $-\frac{k_3 I}{\beta}\left(\frac{D}{y}\right)^{k_3-k_1}$ by setting $f_-(y) = 0$, and substitute this expression into $f'_-(y)$ to give a new function $g(y)$. We will then show that $g(y) < 0$, $\forall y > S_0$, by showing that

$$\begin{aligned}g(S_0) &\leq 0,\\ g'(y) &< 0 \text{ for } y > S_0.\end{aligned}$$

First let's consider the case $\nu_D \neq r$,

$$-\frac{k_3 I}{\beta}\left(\frac{D}{y}\right)^{k_3-k_1} = \frac{k_1 S_0}{r} + \frac{(k_1-1)y}{\nu_D - r} + (k_2-k_1)B_0 y^{k_2}. \tag{5.26}$$

Taking derivatives and substituting (5.26) into $f'_-(y)$,

$$\begin{aligned}g(y) &= \frac{(k_1-1)}{\nu_D-r} + k_2(k_2-k_1)B_0 y^{k_2-1} + -(k_3-k_1)\frac{k_3 I}{\beta}D^{k_3-k_1}y^{-k_3+k_1-1}\\ &= \frac{(k_1-1)}{\nu_D-r} + k_2(k_2-k_1)B_0 y^{k_2-1}\\ &\quad +(k_3-k_1)\left\{\frac{k_1 S_0}{ry} + \frac{(k_1-1)}{\nu_D-r} + (k_2-k_1)B_0 y^{k_2-1}\right\}\\ &= (k_3-k_1)\frac{k_1 S_0}{ry} + (k_3-k_1+1)\frac{(k_1-1)}{\nu_D-r} + (k_3-k_1+k_2)(k_2-k_1)B_0 y^{k_2-1}\\ &= \frac{k_1(k_3-k_1)}{r}\frac{S_0}{y} + (k_3-k_1+1)\frac{(k_1-1)}{\nu_D-r}\\ &\quad +(k_3-k_1+k_2)(k_2-k_1)\frac{r-\nu_D k_1}{r(\nu_D-r)(k_2-k_1)S_0^{k_2-1}}y^{k_2-1}\\ &= \frac{k_1(k_3-k_1)}{r}\frac{S_0}{y} + (k_3-k_1+1)\frac{(k_1-1)}{\nu_D-r} + \frac{(k_3-k_1+k_2)(r-\nu_D k_1)}{r(\nu_D-r)}\left(\frac{y}{S_0}\right)^{k_2-1}.\end{aligned}$$

At $y = S_0$, we have

$$\begin{aligned}
g(S_0) &= \frac{k_1(k_3-k_1)}{r}\frac{S_0}{S_0} + (k_3-k_1+1)\frac{(k_1-1)}{\nu_D - r} + \frac{(k_3-k_1+k_2)(r-\nu_D k_1)}{r(\nu_D - r)}\left(\frac{S_0}{S_0}\right)^{k_2-1} \\
&= \frac{k_1(k_3-k_1)}{r} + (k_3-k_1+1)\frac{(k_1-1)}{\nu_D - r} + \frac{(k_3-k_1+k_2)(r-\nu_D k_1)}{r(\nu_D - r)} \\
&= \frac{k_1(k_3-k_1)(\nu_D - r) + r(k_1-1)(k_3-k_1+1) + (k_3-k_1+k_2)(r-\nu_D k_1)}{r(\nu_D - r)} \\
&= \frac{\nu_D k_1(k_3-k_1) - rk_1(k_3-k_1) + rk_1(k_3-k_1) + rk_1 - r(k_3-k_1) - r}{r(\nu_D - r)} \\
&\quad + \frac{r(k_3-k_1) + rk_2 - \nu_D k_1(k_3-k_1) - \nu_D k_1 k_2}{r(\nu_D - r)} \\
&= \frac{r(k_1+k_2-1) - \nu_D k_1 k_2}{r(\nu_D - r)} = 0.
\end{aligned}$$

Taking derivatives

$$\begin{aligned}
g'(y) &= \frac{-k_1(k_3-k_1)}{r}\frac{S_0}{y^2} + \frac{(k_2-1)(k_3-k_1+k_2)(r-\nu_D k_1)}{r(\nu_D - r)}\left(\frac{y^{k_2-2}}{S_0^{k_2-1}}\right) \\
&= \frac{S_0}{ry^2}\left\{-k_1(k_3-k_1) + \frac{(k_2-1)(k_3-k_1+k_2)(r-\nu_D k_1)}{\nu_D - r}\left(\frac{y}{S_0}\right)^{k_2}\right\}.
\end{aligned}$$

The first expression $-k_1(k_3-k_1)$ is negative because $k_1 > 0$ and $k_3 > k_1$. We note that $k_2 - 1 < 0$ and $(r - \nu_D k_1)/(\nu_D - r) < 0$. Thus the second term has the same sign as $k_3 - k_1 + k_2$. If $k_3 - k_1 + k_2 < 0$, both terms are negative and we have established that $g'(y) < 0$. Now let's consider the case where $k_3 - k_1 + k_2 > 0$. Since the second term is positive and $(\frac{y}{S_0})^{k_2} < 1$,

$$g'(y) < \frac{S_0}{ry^2}\left\{-k_1(k_3-k_1) + \frac{(k_2-1)(k_3-k_1+k_2)(r-\nu_D k_1)}{\nu_D - r}\right\}.$$

We can split our positive term into a positive and negative part

$$\frac{(k_2-1)(k_3-k_1+k_2)(r-\nu_D k_1)}{\nu_D - r} = \frac{(k_2-1)(k_3-k_1)(r-\nu_D k_1)}{\nu_D - r} + \frac{(k_2-1)k_2(r-\nu_D k_1)}{\nu_D - r}.$$

Since $k_2 < 0$, the second term is negative and

$$\begin{aligned} g'(y) &< \frac{S_0}{ry^2}\left\{-k_1(k_3-k_1) + \frac{(k_2-1)(k_3-k_1)(r-\nu_D k_1)}{\nu_D - r}\right\} \\ &= \frac{S_0(k_3-k_1)}{ry^2(\nu_D - r)}\{-k_1(\nu_D - r) + (k_2-1)(r-\nu_D k_1)\} \\ &= \frac{S_0(k_3-k_1)}{ry^2(\nu_D - r)}\{-k_1\nu_D + rk_1 + rk_2 - r - \nu_D k_1 k_2 + \nu_D k_1\} = 0. \end{aligned}$$

Now consider the case $\nu_D = r$,

$$-\frac{k_3 I}{\beta}\left(\frac{D}{y}\right)^{k_3-1} = \frac{S_0}{r} - \frac{2y}{\sigma_D^2 + 2r} + \left(\frac{-2r}{\sigma_D^2} - 1\right) B_0 y^{\frac{-2r}{\sigma_D^2}}. \tag{5.27}$$

Taking derivatives and substituting (5.27) into our expression for $f_-(y)$,

$$\begin{aligned} g(y) &= -\frac{2}{\sigma_D^2 + 2r} + \frac{-2r}{\sigma_D^2}\left(\frac{-2r}{\sigma_D^2} - 1\right) B_0 y^{\frac{-2r}{\sigma_D^2}-1} - (k_3-1)\frac{k_3 I}{\beta} D^{k_3-1} y^{-k_3+1-1} \\ &= -\frac{2}{\sigma_D^2 + 2r} + \frac{-2r}{\sigma_D^2}\left(\frac{-2r}{\sigma_D^2} - 1\right) B_0 y^{\frac{-2r}{\sigma_D^2}-1} \\ &\quad +(k_3-1)\left\{\frac{S_0}{ry} - \frac{2}{\sigma_D^2 + 2r} + \left(\frac{-2r}{\sigma_D^2} - 1\right) B_0 y^{\frac{-2r}{\sigma_D^2}-1}\right\} \\ &= (k_3-1)\frac{S_0}{ry} - \frac{2k_3}{\sigma_D^2 + 2r} + \left(k_3 - 1 + \frac{-2r}{\sigma_D^2}\right)\left(\frac{-2r}{\sigma_D^2} - 1\right) B_0 y^{\frac{-2r}{\sigma_D^2}-1} \\ &= \frac{(k_3-1)}{r}\frac{S_0}{y} - \frac{2k_3}{\sigma_D^2 + 2r} \\ &\quad + \left(k_3 - 1 + \frac{-2r}{\sigma_D^2}\right)\left(\frac{-2r}{\sigma_D^2} - 1\right) \frac{\frac{1}{2}\sigma_D^2 S^{\frac{2r}{\sigma_D^2}+1}}{r(\frac{1}{2}\sigma_D^2 + r)(\frac{2r}{\sigma_D^2} + 1)} y^{\frac{-2r}{\sigma_D^2}-1} \\ &= \frac{(k_3-1)}{r}\frac{S_0}{y} - \frac{2k_3}{\sigma_D^2 + 2r} + \frac{2r - \sigma_D^2(k_3-1)}{r(\sigma_D^2 + 2r)}\left(\frac{S_0}{y}\right)^{\frac{2r}{\sigma_D^2}+1}. \end{aligned}$$

At $y = S_0$, we have

$$\begin{aligned} g(S_0) &= \frac{(k_3-1)}{r}\frac{S_0}{S_0} - \frac{2k_3}{\sigma_D^2 + 2r} + \frac{2r - \sigma_D^2(k_3-1)}{r(\sigma_D^2 + 2r)}\left(\frac{S_0}{S_0}\right)^{\frac{2r}{\sigma_D^2}+1} \\ &= \frac{(k_3-1)}{r} - \frac{2k_3}{\sigma_D^2 + 2r} + \frac{2r - \sigma_D^2(k_3-1)}{r(\sigma_D^2 + 2r)} \\ &= \frac{(k_3-1)(\sigma_D^2 + 2r) - 2rk_3 + 2r - \sigma_D^2(k_3-1)}{r(\sigma_D^2 + 2r)} \\ &= \frac{\sigma_D^2 k_3 - \sigma_D^2 + 2rk_3 - 2r - 2rk_3 + 2r - \sigma_D^2 k_3 + \sigma_D^2}{r(\sigma_D^2 + 2r)} = 0. \end{aligned}$$

Taking derivatives

$$\begin{aligned} g'(y) &= \frac{-(k_3-1)}{r}\frac{S_0}{y^2} + \left(\frac{-2r}{\sigma_D^2} - 1\right)\frac{(2r - \sigma_D^2(k_3-1))}{ry(\sigma_D^2+2r)}\left(\frac{S_0}{y}\right)^{\frac{2r}{\sigma_D^2}+1} \\ &= \frac{S_0}{ry^2}\left\{-(k_3-1) + \frac{\sigma_D^2(k_3-1)-2r}{\sigma_D^2}\left(\frac{S_0}{y}\right)^{\frac{2r}{\sigma_D^2}}\right\}. \end{aligned}$$

The first expression $-(k_3-1)$ is negative because $k_3 > k_1 = 1$. The second term may be positive or negative depending on the value of k_3. If the second term is negative, both terms are negative and we have established that $g'(y) < 0$. Now let's consider the case where the second term is positive. Since $(\frac{y}{S_0})^{k_2} < 1$,

$$\begin{aligned} g'(y) &< \frac{S_0}{ry^2}\left\{-(k_3-1) + \frac{\sigma_D^2(k_3-1)-2r}{\sigma_D^2}\right\} \\ &= \frac{-2r}{\sigma_D^2} < 0. \end{aligned}$$

□

Lemma 5.13. *For $y \in [S_1, \infty)$, we can write*

$$F'(D,y) = \frac{\beta}{y}\left(\frac{D}{y}\right)^{k_1} f_+(y),$$

where

$$f_+(y) = \frac{k_1(S_0-S_1)}{r} + (k_2-k_1)(B_0-B_1)y^{k_2} + \frac{k_3 I}{\beta}\left(\frac{D}{y}\right)^{k_3-k_1}.$$

Proof. In each case we have,

$$\begin{aligned} F(D,y) &= \beta\left(\frac{-S_0}{r} + B_0 y^{k_2}\right)\left(\frac{D}{y}\right)^{k_1} - \beta\phi_0(D) + \beta C \\ &\quad -\beta\left(\frac{-S_1}{r} + B_1 y^{k_2}\right)\left(\frac{D}{y}\right)^{k_1} + \beta\phi_1(D) - I\left(\frac{D}{y}\right)^{k_3} \\ &= \beta(-\phi_0(D) + C + \phi_1(D)) \\ &\quad +\beta D^{k_1}\left(\frac{-(S_0-S_1)}{r}y^{-k_1} + (B_0-B_1)y^{k_2-k_1}\right) \\ &\quad -I\left(\frac{D}{y}\right)^{k_3}. \end{aligned}$$

Thus

$$\begin{aligned}\frac{\partial F(D,y)}{\partial y} &= \frac{\beta}{y}\left(\frac{D}{y}\right)^{k_1}\left(\frac{k_1(S_0-S_1)}{r}+(k_2-k_1)(B_0-B_1)y^{k_2}\right)\\ &+k_3ID^{k_3}\left(\frac{1}{y}\right)^{k_3+1}\\ &= \frac{\beta}{y}\left(\frac{D}{y}\right)^{k_1} f_+(y).\end{aligned}$$

□

Lemma 5.14. $f'_+(y) < 0 \ \forall y \in (S_1, \infty)$

Proof. Taking derivatives

$$f'_+(y) = k_2(k_2-k_1)(B_0-B_1)y^{k_2} - (k_3-k_1)\frac{k_3 I}{\beta} D^{k_3-k_1} y^{-k_3+k_1-1}.$$

Since $B_0 - B_1 < 0$ and $k_3 - k_1 > 0$, both expressions are negative and so

$$f'_+(y) < 0.$$

□

Lemma 5.15.

$$\lim_{y\to\infty} f_+(y) = \frac{k_1(S_0-S_1)}{r} < 0.$$

Proof. Since $k_2 < 0$ and $-(k_3-k_1) < 0$ the expressions containing y^{k_2} and $y^{-(k_3-k_1)}$ vanish as $y \to \infty$. □

Theorem 5.16. *If $f_-(S_1) \leq 0$ then the optimal trigger lies in the second region $(S_0, S_1]$ and y^* is the unique root of $f_-(y) = 0$. Otherwise, the optimal trigger lies in the third region $[S_1, \infty)$ and y^* is the unique root of $f_+(y) = 0$.*

Proof. From (5.23) we know that $f_-(S_0) > 0$. Suppose that $f_-(S_1) \leq 0$ then there exists some value $y_1 \in (S_0, S_1)$ with $f_-(y_1) = 0$. Since $f_-(y) = 0$ implies $f'_-(y) < 0$ (Lemma 5.12), we know that this zero is unique. Since $f_+(S_1) = f_-(S_1) < 0$ and

$f_+(y)$ is a decreasing function, there are no zeros for $f_+(y)$ on (S_1, ∞). Thus y_1 is the unique optimal trigger. Now suppose that $f_-(S_1) > 0$. Lemma 5.12 implies that there are no zeros for $f_-(y)$ on (S_0, S_1). Since $f_+(S_1) = f_-(S_1) > 0$ and $f_+(y)$ is a decreasing function (Lemma 5.14) and converges to a negative number (Lemma 5.15), there exists some value $y_2 \in (S_1, \infty)$ for which $f_+(y_2) = 0$. Thus y_2 is the unique optimal trigger. □

Algorithm 5.17. *If $f_-(S_1) < 0$, then apply the bisection method to $f_-(y)$ on $[S_0, S_1]$. Otherwise, apply the bisection method to $f_+(y)$ on $[S_1, y_+]$ where*

$$y_+ = \max\left(\left[\frac{k_1(S_1 - S_0)}{3r(k_2 - k_1)(B_0 - B_1)}\right]^{1/k_2}, D\left[\frac{\beta k_1(S_1 - S_0)}{3rk_3 I}\right]^{-1/(k_3 - k_1)}\right).$$

Proof. This algorithm comes directly from Theorem 5.16. We need only find an end value y_+ which satisfies $f_+(y_+) < 0$. Choose y_1 such that

$$(k_2 - k_1)(B_0 - B_1)y_1^{k_2} < \frac{k_1(S_1 - S_0)}{3r},$$

i.e.

$$y_1 = \left[\frac{k_1(S_1 - S_0)}{3r(k_2 - k_1)(B_0 - B_1)}\right]^{1/k_2}.$$

Next, we choose y_2 such that

$$\frac{k_3 I}{\beta}\left(\frac{D}{y_2}\right)^{k_3 - k_1} < \frac{k_1(S_1 - S_0)}{3r},$$

i.e.

$$y_2 = D\left[\frac{\beta k_1(S_1 - S_0)}{3rk_3 I}\right]^{-1/(k_3 - k_1)}.$$

Finally, choose $y^+ = \max(y_1, y_2)$ then

$$\begin{aligned} f_+(y_+) &= \frac{k_1(S_0 - S_1)}{r} + (k_2 - k_1)(B_0 - B_1)y_+^{k_2} + \frac{k_3 I}{\beta}\left(\frac{D}{y_+}\right)^{k_3 - k_1} \\ &< \frac{k_1(S_0 - S_1)}{3r} < 0. \end{aligned}$$

□

5.3.6 Numerical Examples

This section provides some numerical examples for the $\nu_D \neq r$ and $\nu_D = r$ cases. The $\nu_D \neq r$ examples, labelled 5.1a-d, use the ν_D, σ_D and r parameters in d'Halluin et al. [20]. The $\nu_D = r$ examples, labelled 5.2a-d, use the r and σ parameters in Dixit and Pindyck [22]. As explained in Chapter 4, we assume that costs decrease by a factor of two every 18 months (i.e. $\alpha = 0.462$). For each case, we give an example where the optimal trigger y^* lies in the second and third regions. Next, we explain how our model is used when there is no prior investment (i.e. $S_0 = 0$). Finally, we explain how to shift the trigger between the two feasible regions by changing one of the initial parameters and holding the other parameters constant.

As discussed previously, the trigger may lie in the second or third region. Table 5.1 includes some examples for both cases. In Examples 5.1a and 5.2a, y^* lies in the second region (see Figure 5.3). In Examples 5.1b and 5.2b, y^* lies in the third region (see Figure 5.4). Note that in each case, y^* was shifted from the second region to the third region by increasing the ratio $I : \beta$ by a sufficient amount.

Example	ν_D	σ_D	r	D	S_0	S_1	I	β	y^*	$F(D, y^*)$
5.1a	0.75	0.95	0.05	1	1	4	1	0.9	1.9819	56.8090
5.1b	0.75	0.95	0.05	1	1	4	2	0.2	4.3544	11.9094
5.2a	0.04	0.2	0.04	1	1	4	5	0.2	1.9803	8.5534
5.2b	0.04	0.2	0.04	1	1	4	200	0.15	4.0515	5.3642

Table 5.1: Shifting the trigger into the third region

Figures 5.3 to 5.5 show investment values and derivatives for selected examples. The upper graphs show the investment values and the lower graphs show their derivatives. The optimal triggers and transmission capacities are indicated on each graph. We used the values of $F(D, y)$ at each step ($\delta y = 0.01$) to validate the derivative function given in Section 5.3.5:

$$\frac{\partial F(D,y)}{\partial y} \approx \frac{F(D, y+\delta y) - F(D,y)}{\delta y} = \frac{F(D, y+0.01) - F(D,y)}{0.01}.$$

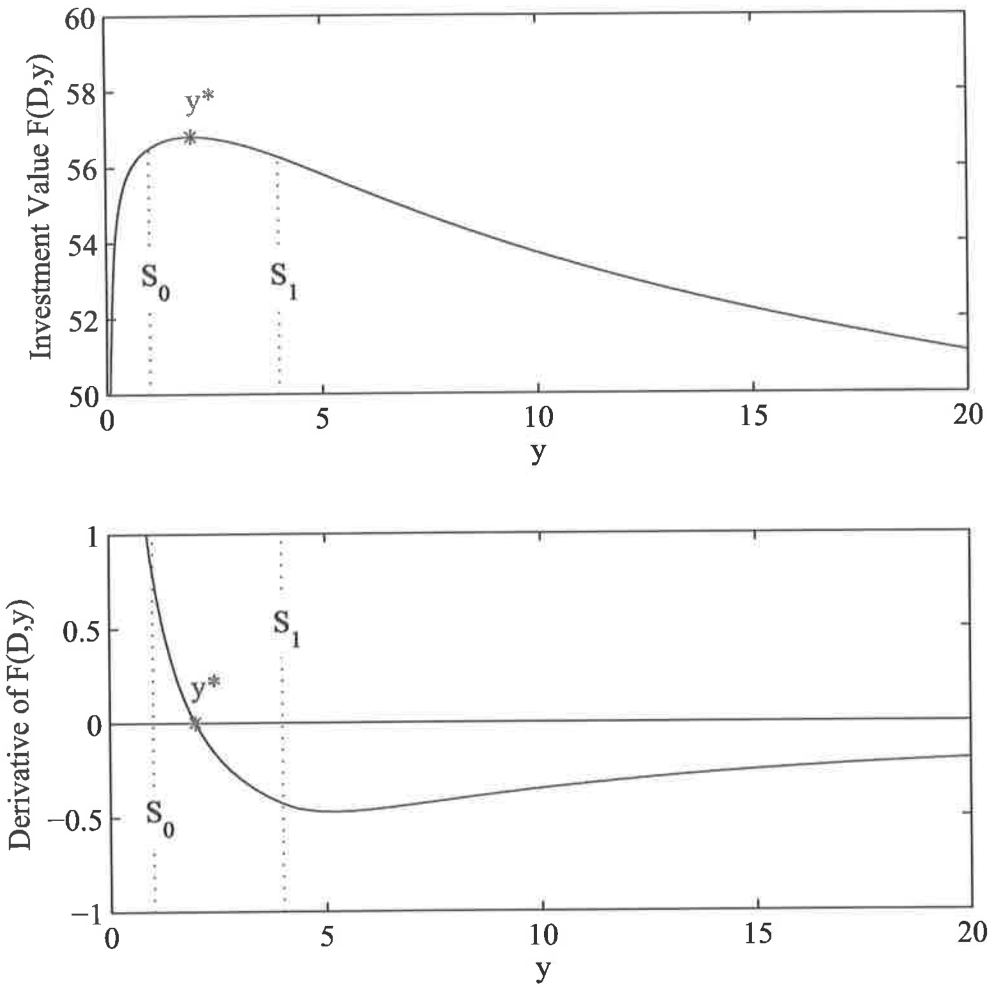


Figure 5.3: Investment values for Example 5.1a

Note that the solid line in Figure 5.5 corresponds to the $\nu_D = r$ model (i.e. Example 5.2c). Two close examples with $r = 0.039$ and $r = 0.041$ are also shown for verification purposes.

The models in Chapters 3 and 4 assume that there is no prior investment. For bandwidth investments, this means that there is no capacity on the link; either no fibre has been laid or no switches have been installed. In the current framework, we

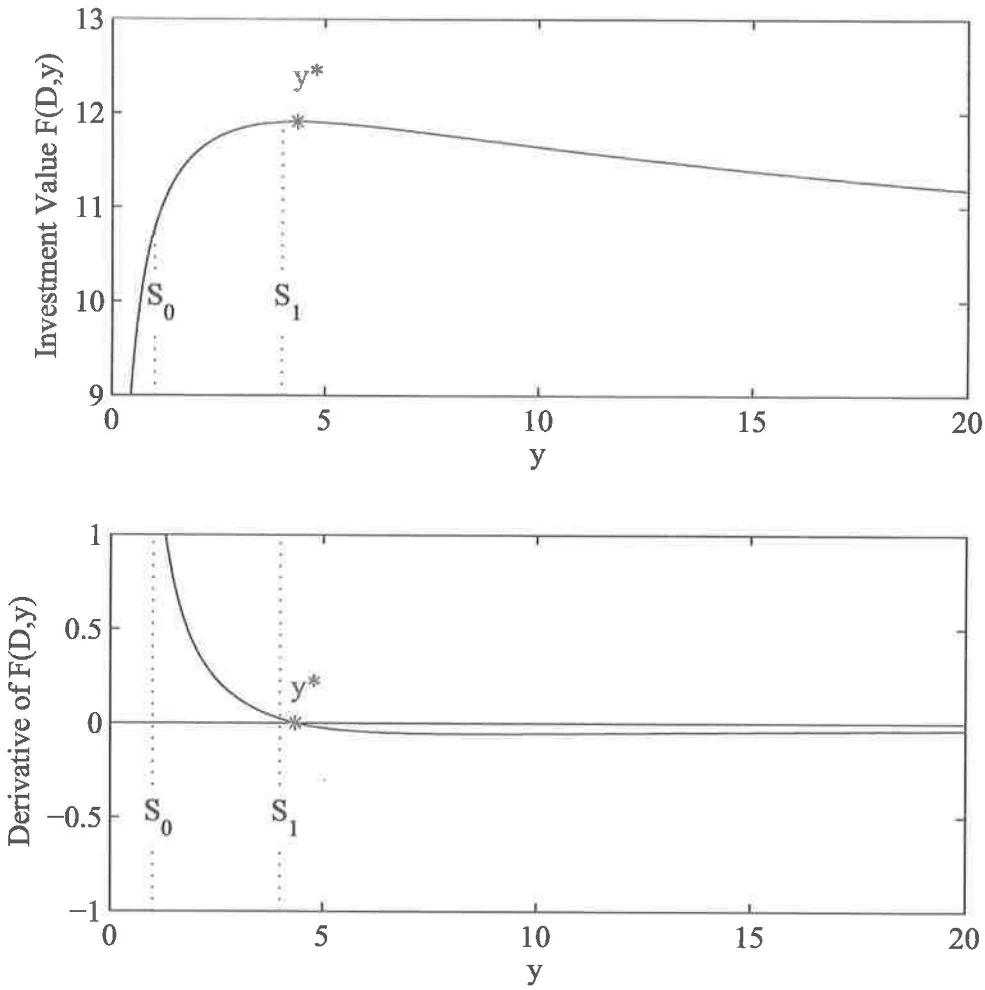


Figure 5.4: Investment values for Example 5.1b

have $S_0 = 0$ and the investment value is

$$F_1(D,y) = \beta \left\{ C - \phi_1(y)E[e^{-r\tau}] + \phi_1(D(0)) \right\} - IE[e^{-(r+\alpha)\tau}],$$

where $C = E\left[\int_0^\infty \min(D(t), S_1)e^{-rt}dt\right]$. We observe that $F(D,y)$ and $F_1(D,y)$ only differ by $\beta\{\phi_0(y)E[e^{-r\tau}] - \phi_0(D(0))\}$,

$$F(D,y) - F_1(D,y) = \beta \left\{ \phi_0(y)E[e^{-r\tau}] - \phi_0(D(0)) \right\}.$$

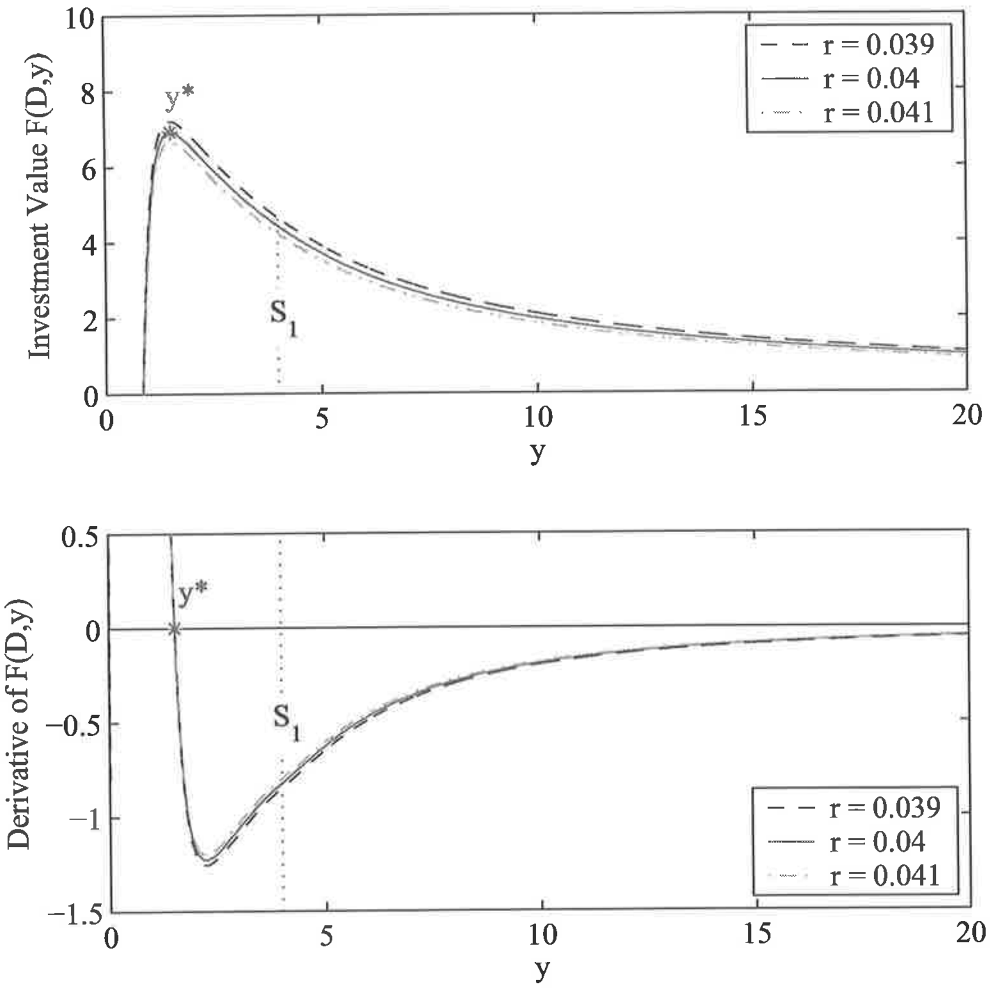


Figure 5.5: Investment values for Example 5.2c

But

$$\phi_0(x) = \begin{cases} \frac{-S_0}{r} + \frac{r - \nu_D k_1}{r(\nu_D - r)(k_2 - k_1) S_0^{k_2 - 1}} x^{k_2}, & \nu_D \neq r; \\ \frac{-S_0}{r} + \frac{r(\ln(S_0)+1)(1-k_1) - \frac{1}{2}\sigma_D^2 k_1}{r(\frac{1}{2}\sigma_D^2 + r)(k_2 - k_1) S_0^{k_2 - 1}} x^{k_2}, & \nu_D = r. \end{cases}$$

and since $k_2 - 1 < 0$, we have $\phi_0(x) = 0$ for all $x > 0$. This means that the two functions are equivalent and so the general formula also works for this special case. We note, however, that only the second and third regions are meaningful. Table 5.2

lists some examples with $S_0 = 0$. We observe that y^* is in the second region $[S_0, S_1]$ in Examples 5.1c and 5.2c. But as before, we can shift y^* into the third region $[S_1, \infty)$ by increasing the ratio of $I : \beta$ (see Examples 5.1d and 5.1d). Example 5.2c is shown in Figure 5.5.

Example	ν_D	σ_D	r	D	S_0	S_1	I	β	y^*	$F(D, y^*)$
5.1c	0.75	0.95	0.05	1	0	4	1	0.9	0.8108	56.5320
5.1d	0.75	0.95	0.05	1	0	4	5	0.2	5.6488	10.5174
5.2c	0.04	0.2	0.04	1	0	4	5	0.2	1.5262	6.9211
5.2d	0.04	0.2	0.04	1	0	4	200	0.1	4.0111	1.8501

Table 5.2: No prior investment

In the previous examples, we shifted y^* between the feasible regions (i.e. $(S_0, S_1]$ and $[S_1, \infty)$) by changing the values of I and β. Theorem 5.16 provides some threshold values for I, β, D, S_0 and S_1. Holding other parameters constant and raising (or lowering) the given parameter beyond its threshold value shifts y^* into the adjacent feasible region. In fact, $y^* \in [S_1, \infty)$ is equivalent to each of the following relations:

$$I \geq \widehat{I}, \tag{5.28}$$

$$\beta \leq \widehat{\beta}, \tag{5.29}$$

$$D \geq \widehat{D}, \tag{5.30}$$

$$S_0 \geq \widehat{S}_0, \tag{5.31}$$

$$S_1 \leq \widehat{S}_1. \tag{5.32}$$

The threshold value for a given parameter is simply the parameter value which makes $g(I, \beta, D, S_0, S_1) = 0$ where

$$g(I, \beta, D, S_0, S_1) = f_-(S_1) = f_+(S_1).$$

Since $f_+(x)$ is identical for the $\nu_D \neq r$ and $\nu_D = r$ cases, we rearranged $f_+(S_1) = 0$

Example	D	S_0	S_1	I	β	y^*	$F(D,y^*)$
5.1a	22.5234	1	4	1	0.9	4	75.7079
5.1a	1	2.8779	4	1	0.9	4	57.1321
5.1a	1	1	1.9819	1	0.9	1.9819	30.1267
5.1a	1	1	4	7.4781	0.9	4	54.0884
5.1a	1	1	4	1	0.1204	4	7.2360
5.1b	0.7507	1	4	2	0.2	4	11.6116
5.1b	1	0.7235	4	2	0.2	4	11.8178
5.1b	1	1	4.3343	2	0.2	4.3343	12.8445
5.1b	1	1	4	1.6618	0.2	4	12.0197
5.1b	1	1	4	2	0.2407	4	14.4656
5.2c	3.4421	0	4	5	0.2	4	12.7682
5.2c	1	3.6267	4	5	0.2	4	9.0547
5.2c	1	0	1.5262	5	0.2	1.5262	3.7094
5.2c	1	0	4	394.9862	0.2	4	3.7094
5.2c	1	0	4	5	0.0025	4	0.0470

Table 5.3: Threshold values

to obtain a general formula for the first three threshold values:

$$\begin{aligned}
\widehat{I} &= \frac{\beta}{k_3}\left(\frac{S_1}{D}\right)^{k_3-k_1}\left\{\frac{k_1(S_1-S_0)}{r}+(k_2-k_1)(B_1-B_0)S_1^{k_2}\right\},\\
\widehat{\beta} &= k_3 I\left(\frac{D}{S_1}\right)^{k_3-k_1}\left\{\frac{k_1(S_1-S_0)}{r}+(k_2-k_1)(B_1-B_0)S_1^{k_2}\right\}^{-1},\\
\widehat{D} &= S_1\left(\frac{\beta}{k_3 I}\left\{\frac{k_1(S_1-S_0)}{r}+(k_2-k_1)(B_1-B_0)S_1^{k_2}\right\}\right)^{1/(k_3-k_1)}.
\end{aligned}$$

The threshold values for the supply levels, $\widehat{S_0}$ and $\widehat{S_1}$, can be found by applying numerical techniques (i.e. the bisection method) to $f_+(S_1) = 0$. Table 5.3 shows the threshold values for Examples 5.1a, 5.1b and 5.2c. In each case we replaced the original value with the threshold value and thereby shifted y^* to the boundary between the second and third regions (i.e. S_1). We note that the $\widehat{S_1} = y^*$ in

Examples 5.1a and 5.2c. This result is generalized in Lemma 5.18. We also note that in some cases it may not be possible to find a value of $\widehat{S}_0$ that lies is within $[0, S_1]$.

Lemma 5.18. *If* $f_-(S_1) < 0$, *then* $\widehat{S}_1 = y^*$.

Proof. Observe that $\widehat{S}_1$ is the unique root of $f_-(y) = 0$. If $f_-(S_1) < 0$, y^* is also a root of $f_-(y) = 0$ and so $\widehat{S}_1 = y^*$.

□

5.4 Conclusion

In this chapter we formulated a general strategy for deciding when to increase the transmission capacity, found an analytical solution for the GBM model and gave some numerical examples. The analysis suggests that there is a unique optimal trigger which is greater than the original transmission capacity. In the next chapter, we will find an analytical solution for an increasing capacity model that incorporates demand saturation. The behaviour of that model will be compared with that of the GBM model.

Chapter 6

Demand Saturation

In the previous chapter we developed a model for increasing link capacity. Like earlier models, this model assumed that the demand process follows a geometric Brownian motion (GBM). However, this assumption raises some questions in a finite population. This chapter extends the increasing link capacity model to support demand saturation.

6.1 Introduction

The previous models assumed that the demand process $D(t)$ followed a GBM. As noted in Chapter 2, a GBM with positive drift will grow exponentially on average, but the demand must be limited by a finite population. Thus we would expect that the demand process will ultimately taper off to some value $\overline{D}$. In Section 6.3 we will use a logistic demand model to model demand saturation

$$dD(t) = \eta(\overline{D} - D(t))D(t)dt + \sigma D(t)dB_D(t).$$

A logarithmic process

$$dD(t) = -\eta \ln \left[\frac{D(t)}{\overline{D}}\right] D(t)dt + \sigma D(t)dB(t),$$

and other mean-reversion processes could also be used. Once the logistic demand model is developed, we would like to compare its behaviour to that of the GBM demand model. We define a logistic variation of the GBM defined by

$$dD(t) = \nu_D D(t)dt + \sigma_D D(t)dB(t),$$

by choosing $\overline{D}$ arbitrarily and then setting $\eta = \frac{\nu_D}{\overline{D}}$ and $\sigma = \sigma_D$. Figure 6.1 shows the sample paths for a GBM and a logistic variation with $\overline{D} = 1000$, LP(1000). We note that the sample paths are close prior to reaching this demand saturation point (i.e. when $D(t) << \overline{D}$), however the GBM continues to grow after the logistic process reaches the demand saturation point.

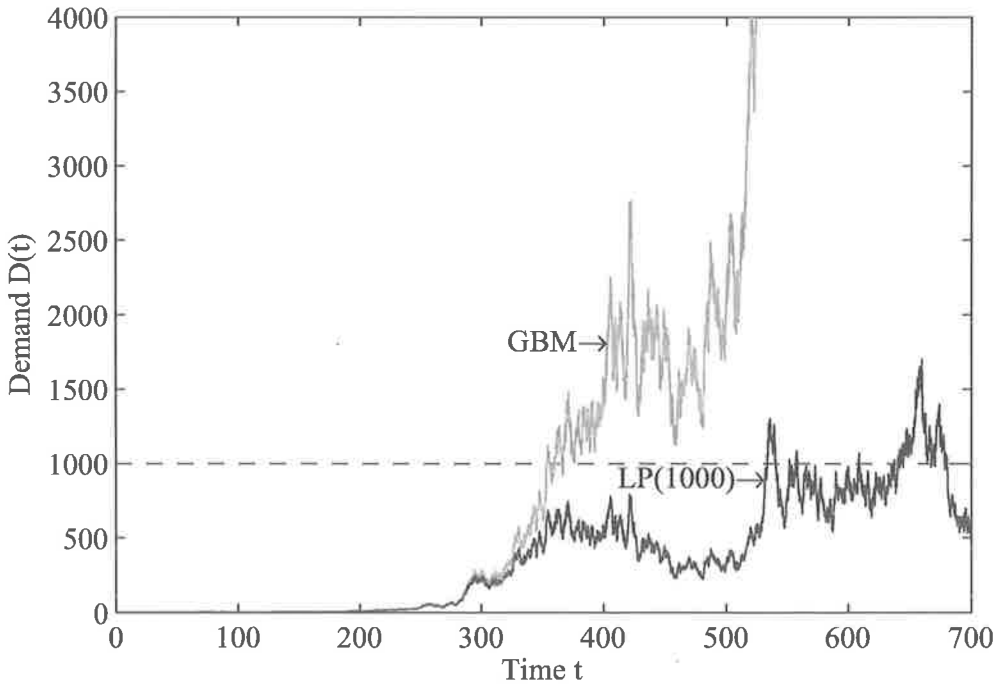


Figure 6.1: A geometric Brownian motion and a logistic variation

6.2 Kummer's Equation

Dixit and Pindyck used a logistic process to extend the classic investment model to mean-reversion models [22, pages 161–167]. Kummer's Equation, which has the form

$$zg''(z) + (b - z)g'(z) - ag(z) = 0,$$

was used to solve their model and it will also be used to solve the logistic model in Section 6.3. Abramowitz and Stegun [1] list eight possible solutions for Kummer's equation. In this thesis we only use three of these solutions

$$\begin{aligned} u_1(z) &= M(a, b, z), \\ u_2(z) &= z^{1-b}M(1 + a - b, 2 - b, z), \\ u_3(z) &= U(a, b, z), \end{aligned}$$

where

$$\begin{aligned} M(a, b, z) &= \sum_{n=0}^{\infty} \frac{(a)_n z^n}{(b)_n n!}, \\ (a)_n &= \Pi_{i=0}^{n}(a + i), \\ U(a, b, z) &= \frac{\pi}{\sin \pi b}\left[\frac{M(a, b, z)}{\Gamma(1 + a - b)\Gamma(b)} - z^{1-b}\frac{M(1 + a - b, 2 - b, z)}{\Gamma(a)\Gamma(2 - b)}\right]. \end{aligned} \quad (6.1)$$

The Wronskians are as follows

$$W(1, 2) = (1 - b)z^{-b}e^z, \quad (6.2)$$

$$W(1, 3) = -\Gamma(b)z^{-b}e^z/\Gamma(a), \quad (6.3)$$

$$W(2, 3) = -\Gamma(2 - b)z^{-b}e^z/\Gamma(1 + a - b). \quad (6.4)$$

We note that $W(1, 2)$ is non-zero when $b \neq 1$. This property ensures that $u_1(z)$ and $u_2(z)$ are linearly independent. The Kummer functions have the following limits

$$\lim_{z\to\infty} M(a, b, z) = \begin{cases} \frac{\Gamma(b)}{\Gamma(b-a)}(-z)^{-a}[1 + O(|z|^{-1})], & z < 0; \\ \frac{\Gamma(b)}{\Gamma(a)}e^z z^{a-b}[1 + O(|z|^{-1})], & z > 0. \end{cases} \quad (6.5)$$

$$\lim_{z\to\infty} U(a, b, z) = z^{-a}[1 + O(|z|^{-1})]. \quad (6.6)$$

where $\Gamma(\cdot)$ is the gamma function defined in Definition 6.1. We shall use the following equations from [1] in subsequent calculations:

13.1.27 $M(a, b, z) = e^z M(b - a, b, -z)$,

13.1.28 $z^{1-b}M(1+a-b,2-b,z) = z^{1-b}e^{z}M(1-a,2-b,-z)$,

13.4.08 $M'(a,b,z) = \frac{a}{b}M(a+1,b+1,z)$,

13.4.10 $aM(a+1,b,z) = aM(a,b,z) + zM(a,b,z)$,

13.4.13 $(b-1)M(a,b-1,z) = (b-1)M(a,b,z) + zM(a,b,z)$.

13.4.23 $a(1+a-b)U(a+1,b,z) = aU(a,b,z) + zU'(a,b,z)$.

Definition 6.1 (Gamma Function). The Gamma function $\Gamma(z)$ is defined by

$$\int_{\infty}^{0} t^{z-1}e^{-t}dt \ (\mathbb{R}z > 0),$$

and has the following properties [1, Chapter 6]:

$$\Gamma(z+1) = z\Gamma(z), \tag{6.7}$$

$$\Gamma(n+1) = n! \text{ if } n \in \mathbb{Z}^{+}. \tag{6.8}$$

6.3 A Logistic Model

The demand process $D(t)$ is assumed to follow a logistic process

$$dD(t) = \eta(\overline{D} - D(t))D(t)dt + \sigma D(t)dB(t),$$

where η is the speed of reversion, $\overline{D}$ is the long-run equilibrium level, σ is the volatility and $B(t)$ is a standard Brownian motion. We observe that $\nu_D > r$ (where ν_D is the drift) for both GBM examples in the literature [41, 20]. Since the logistic variation of a GBM has the property that $\eta\overline{D} = \nu_D$, we will assume that $\eta\overline{D} > r$. Sections 6.3.1- 6.3.4 are used to derive an upper bound for the investment value

$$G(D,y) = \beta\left\{\phi_0(y)E[e^{-r\tau}] - \phi_0(D) + \frac{S_1}{r} - \phi_1(y)E[e^{-r\tau}] + \phi_1(D)\right\} - IE[e^{-(r+\alpha)\tau}],$$

where $\tau = \inf\{t \geq 0 : D(t) \geq y\}$. Note that $G(D, y)$ comes directly from Theorem 5.3 and the property that

$$C \quad E\left[\int_0^\infty \min[D(s), S_1]e^{-rs}ds\right] \leq \frac{S_1}{r}.$$

Section 6.3.5 discusses whether the optimal trigger can be found. Finally, some numerical examples are presented in Section 6.3.6. Note that the following ψ functions and some results in Appendix B.5 are used to simplify calculations:

$$\begin{aligned}
\psi_1(x) &= x^a M\left(a, b, \frac{2\eta x}{\sigma^2}\right), \\
\psi_2(x) &= x^{a+1-b} M\left(a+1-b, 2-b, \frac{2\eta x}{\sigma^2}\right), \\
\psi_3(x) &= x^{a_1} M\left(a_1, b_1, \frac{2\eta x}{\sigma^2}\right), \\
\psi_4(x) &= x^a U\left(a, b, \frac{2\eta x}{\sigma^2}\right), \\
\psi_5(x) &= M\left(a, b, \frac{2\eta x}{\sigma^2}\right), \\
\psi_6(x) &= \left(\frac{2\eta x}{\sigma^2}\right)^{1-b} M\left(a+1-b, 2-b, \frac{2\eta x}{\sigma^2}\right), \\
\psi_7(x) &= \left(\frac{2\eta x}{\sigma^2}\right)^{1-a} Y\left(1-a, 2-b, \frac{-2\eta x}{\sigma^2}\right), \\
\psi_8(x) &= \left(\frac{2\eta x}{\sigma^2}\right)^{b-a} Y\left(b-a, b, \frac{-2\eta x}{\sigma^2}\right), \\
\psi_9(x) &= \frac{\left(\frac{2\eta x}{\sigma^2}\right)^{-a} M\left(-a, 2-b, \frac{-2\eta x}{\sigma^2}\right)}{-a}, \\
\psi_{10}(x) &= \frac{\left(\frac{2\eta x}{\sigma^2}\right)^{b-a-1} M\left(b-a-1, b, \frac{-2\eta x}{\sigma^2}\right)}{b-a-1},
\end{aligned}$$

where

$$\begin{aligned}
a &= \frac{-\left[\eta\overline{D}-\frac{\sigma^2}{2}\right]+\sqrt{\left(\eta\overline{D}-\frac{\sigma^2}{2}\right)^2+2r\sigma^2}}{\sigma^2},\\
b &= \frac{2\overline{D}\eta}{\sigma^2}+2a,\\
a_1 &= \frac{-\left[\eta\overline{D}-\frac{\sigma^2}{2}\right]+\sqrt{\left(\eta\overline{D}-\frac{\sigma^2}{2}\right)^2+2(r+\alpha)\sigma^2}}{\sigma^2},\\
b_1 &= \frac{2\overline{D}\eta}{\sigma^2}+2a_1.
\end{aligned}$$

6.3.1 An expression for $\mathcal{L}_\lambda\phi(x)$

Lemma 6.1 provides an expression for $\mathcal{L}_\lambda\phi$ which satisfies (5.1) when the demand process follows the logistic process.

Lemma 6.1. *If the demand process $D(t)$ follows a logistic process*

$$dD(t) = \eta(\overline{D}-D(t))D(t)dt + \sigma D(t)dB(t),$$

then $\mathcal{L}_\lambda\phi = \frac{1}{2}\sigma_D^2x^2\phi''(x) + \eta(\overline{D}-x)x\phi'(x) - \lambda\phi$ satisfies (5.1).

Proof. By Itô's lemma, we have for any C^2 function ϕ

$$\begin{aligned}
d[\phi(D(t))e^{-\lambda t}] &= -\lambda\phi(D(t))e^{-\lambda t}dt + e^{-\lambda t}\phi'(D(t))dD(t) + e^{-\lambda t}\phi''(D(t))\frac{1}{2}dD(t)^2\\
&= -\lambda\phi(D(t))e^{-\lambda t}dt + e^{-\lambda t}\phi'(D(t))[\eta(\overline{D}-D(t))D(t)dt + \sigma_D D(t)dB(t)]\\
&\quad +e^{-\lambda t}\phi''(D(t))\frac{\sigma_D^2}{2}D(t)^2dt\\
&= e^{-\lambda t}\left\{-\lambda\phi(D(t)) + \eta(\overline{D}-D(t))D(t)\phi(D(t))' + \frac{\sigma_D^2}{2}D(t)^2\phi''(D(t))\right\}dt\\
&\quad +\sigma_D e^{-\lambda t}\phi'(D(t))D(t)dB(t).
\end{aligned}$$

Let $\mathcal{L}_\lambda\phi(x) = -\lambda\phi(x) + \eta(\overline{D}-x)x\phi'(x) + \frac{1}{2}\sigma_D^2x^2\phi''(x)$

$$d[\phi(D(t))e^{-\lambda t}] = \mathcal{L}\phi(D(s))e^{-\lambda t}dt + \sigma_D\phi'(D(t))D(t)e^{-\lambda t}dB(t).$$

□

6.3.2 A general solution for $\mathcal{L}_\lambda\phi(x) = 0$

Lemma 6.2 uses Kummer's equation to find a general solution for $\mathcal{L}_\lambda\phi(x) = 0$.

Lemma 6.2. *The general solution of*

$$\mathcal{L}_\lambda\phi(x) = 0,$$

has the following form

$$\phi(x) = Ax^a M\left(a, b, \frac{2\eta x}{\sigma^2}\right) + Bx^{a+1-b} M\left(a+1-b, 2-b, \frac{2\eta x}{\sigma^2}\right),$$

where a is the positive root of

$$\frac{1}{2}\sigma^2 a(a-1) + \eta\overline{D}a - \lambda = 0, \tag{6.9}$$

and $b = \frac{2\overline{D}\eta}{\sigma^2} + 2a$.

Proof. We seek a solution to

$$\mathcal{L}_\lambda\phi(x) = \frac{1}{2}\sigma^2 x^2\phi''(x) + \eta(\overline{D} - x)x\phi'(x) - \lambda\phi(x) = 0. \tag{6.10}$$

If $\phi(x) = Ax^\theta h(x)$ then

$$\begin{aligned}
\phi(x) &= Ax^\theta h(x), \\
\phi'(x) &= A\theta x^{\theta-1}h(x) + Ax^\theta h'(x), \\
\phi''(x) &= A\theta(\theta-1)x^{\theta-2}h(x) + 2A\theta x^{\theta-1}h'(x) + Ax^\theta h''(x).
\end{aligned}$$

Substituting these expressions into (6.10) we get

$$\begin{aligned}
0 &= \frac{1}{2}\sigma x^2\left[A\theta(\theta-1)x^{\theta-2}h + 2A\theta x^{\theta-1}h'(x) + Ax^\theta h''(x)\right] \\
&\quad +\eta(\overline{D} - x)x\left[A\theta x^{\theta-1}h(x) + Ax^\theta h'(x)\right] - \lambda Ax^\theta h(x) \\
&= h\left[\frac{1}{2}\sigma^2\theta(\theta-1) + \eta\overline{D}\theta - \lambda\right]x^\theta + \frac{1}{2}2\sigma^2\theta x^{\theta+1}h'(x) + \frac{1}{2}\sigma^2 x^{\theta+2}h''(x) \\
&\quad -\eta\theta x^{\theta+1}h(x) + \eta(\overline{D} - x)x^{\theta+1}h'(x).
\end{aligned}$$

Now choose θ such that

$$\frac{1}{2}\sigma^2\theta(\theta-1) + \eta\overline{D}\theta - \lambda = 0,$$

then

$$-\eta\theta h(x) + \left[\eta(\overline{D} - x) + \sigma^2\theta\right] h'(x) + \frac{1}{2}\sigma^2 x h''(x) = 0.$$

By making the substitution $z = \frac{2\eta x}{\sigma^2}$

$$\begin{aligned} h(x) &= g(z), \\ h'(x) &= g'(z)\frac{2\eta}{\sigma^2}, \\ h''(x) &= g''(z)\left(\frac{2\eta}{\sigma^2}\right)^2, \end{aligned}$$

So

$$-\eta\theta g(z) + \left[\eta\overline{D} - \eta\frac{\sigma^2 z}{2\eta} + \sigma^2\theta\right]\frac{2\eta}{\sigma^2}g'(z) + \frac{\sigma^2}{2}\frac{\sigma^2}{2\eta}zg''(z)\left(\frac{2\eta}{\sigma^2}\right)^2 = 0.$$

If we let $a = \theta$, $b = \frac{2\overline{D}\eta}{\sigma^2} + 2a$, then

$$\eta\left\{zg''(z) + (b - z)g'(z) - ag(z)\right\} = 0.$$

The equation

$$zg''(z) + (b - z)g'(z) - ag(z) = 0, \tag{6.11}$$

is known as Kummer's Equation. Abramowitz and Stegan [1] list eight possible solutions. If $b \neq 1$, then

$$\begin{aligned} u_1(z) &= M(a, b, z), \\ u_2(z) &= z^{1-b}M(1 + a - b, 2 - b, z) \end{aligned}$$

are linearly independent (since the Wronskian is zero [1]) and

$$g(z) = A_1 M(a, b, z) + B_1 z^{1-b} M(1 + a - b, 2 - b, z)$$

is a general solution of (6.11). Thus

$$\begin{aligned} \phi(x) &= x^a h(x) = x^a\left\{A_1 M\left(a, b, \frac{2\eta x}{\sigma^2}\right) + B_1\left(\frac{2\eta x}{\sigma^2}\right)^{1-b} M\left(1 + a - b, 2 - b, \frac{2\eta x}{\sigma^2}\right)\right\} \\ &= A_1 x^a M\left(a, b, \frac{2\eta x}{\sigma^2}\right) + B_1\left(\frac{2\eta}{\sigma^2}\right)^{1-b} x^{a+1-b} M\left(a + 1 - b, 2 - b, \frac{2\eta x}{\sigma^2}\right). \end{aligned}$$

Since $\left(\frac{\sigma^2}{2\eta}\right)^{1-b}$ is a constant we have

$$\phi(x) = Ax^a M\left(a, b, \frac{2\eta x}{\sigma^2}\right) + Bx^{a+1-b} M\left(a+1-b, 2-b, \frac{2\eta x}{\sigma^2}\right),$$

where $A_1 = A$ and $B = B_1\left(\frac{\sigma^2}{2\eta}\right)^{1-b}$. □

6.3.3 A unique bounded solution for $\mathcal{L}_r\phi(x) = \min(x, S)$

Lemma 6.3 provides a unique bounded solution for $\mathcal{L}_r\phi(x) = \min(x, S)$. Henceforth, we shall let $\phi_0(x)$ and $\phi_1(x)$ denote the solutions for $S = S_0$ and $S = S_1$ respectively. Figure 6.2 shows the ϕ functions for selected logistic versions of Example 5.1a. As expected, the ϕ functions converge to that of the GBM model as $\overline{D} \to \infty$.

Lemma 6.3. *The unique bounded solution of $\mathcal{L}_r\phi(x) = \min(x, S)$ is:*

$$\phi(x) = \begin{cases} A\psi_1(x) + \frac{2\Phi(x)}{\sigma^2(b-1)}, & 0 \le x \le S; \\ B\psi_4(x) - \frac{S}{r}, & x \ge S, \end{cases}$$

where

$$\begin{aligned}
\psi_1(x) &= x^a M\left(a, b, \frac{2\eta x}{\sigma^2}\right), \\
\psi_4(x) &= x^a U\left(a, b, \frac{2\eta x}{\sigma^2}\right), \\
\Phi(x) &= xM\left(1+a-b, 2-b, \frac{2\eta x}{\sigma^2}\right) Y\left(b-a, b, \frac{-2\eta x}{\sigma^2}\right) \\
&\quad -xM\left(a, b, \frac{2\eta x}{\sigma^2}\right) Y\left(1-a, 2-b, \frac{-2\eta x}{\sigma^2}\right), \\
a &= \frac{-\left[\eta\overline{D} - \frac{\sigma^2}{2}\right] + \sqrt{\left(\eta\overline{D} - \frac{\sigma^2}{2}\right)^2 + 2r\sigma^2}}{\sigma^2}, \\
b &= \frac{2\overline{D}\eta}{\sigma^2} + 2a, \\
Y(a, b, z) &= \sum_{n=0}^{\infty} \frac{(a)_n}{(b)_n} \frac{z^n}{(a+n)n!}.
\end{aligned}$$

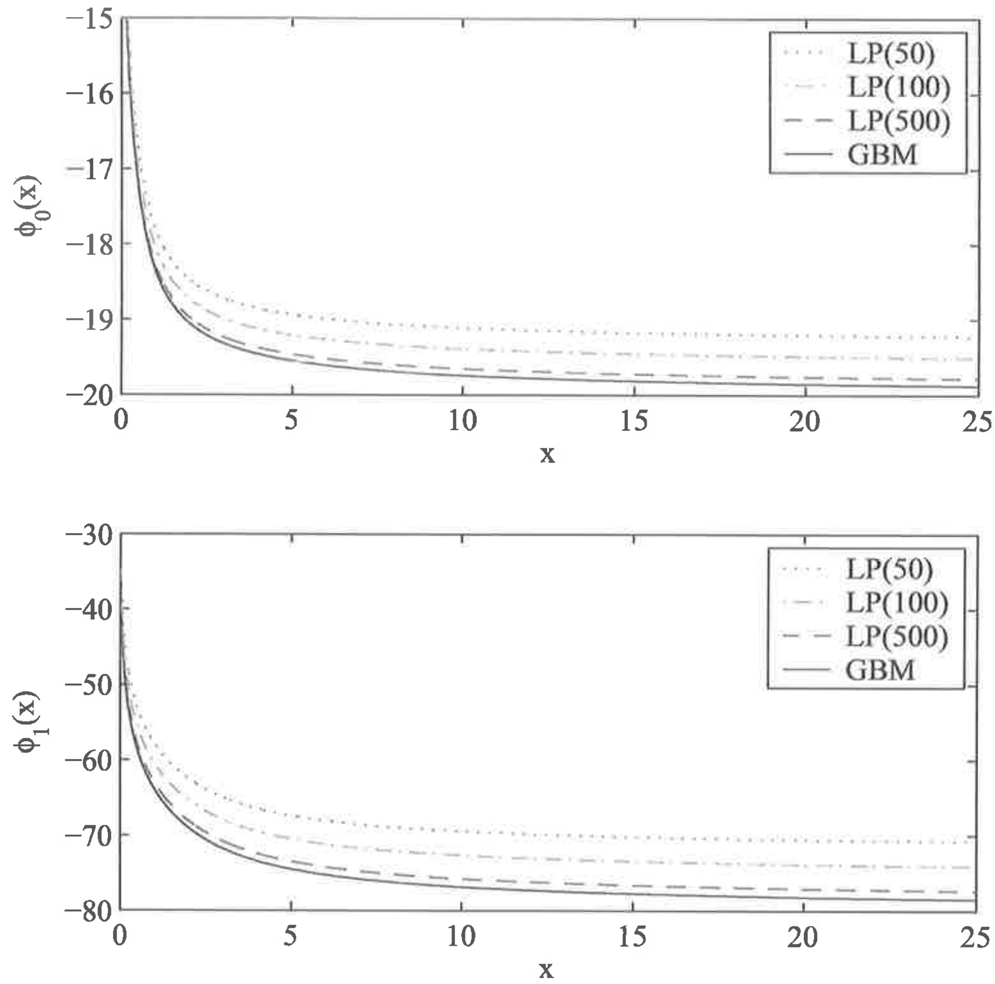


Figure 6.2: The ϕ functions for logistic versions of Example 5.1a

Proof. We seek a unique, bounded solution of

$$\mathcal{L}_r\phi(x) = \min(x, S),$$

with $|x\phi'(x)|$ bounded. Let's consider $x \leq S$ first

$$\mathcal{L}_r\phi(x) = x.$$

By Lemma 6.2, the general solution of $\mathcal{L}_r\phi = 0$ is

$$\phi(x) = Ax^a M\left(a, b, \frac{2\eta x}{\sigma^2}\right) + Bx^{a+1-b} M\left(a+1-b, 2-b, \frac{2\eta x}{\sigma^2}\right),$$

where a is the positive root of

$$\frac{1}{2}\sigma^2 a(a-1) + \eta\overline{D}a - r = 0,$$

and $b = \frac{2\overline{D}\eta}{\sigma^2} + 2a$. In the proof of this lemma we observed that $\mathcal{L}\phi = 0$ can be reduced to the Kummer equation

$$\eta\left\{zg''(z) + (b-z)g'(z) - ag(z)\right\} = 0, \qquad (6.12)$$

and chose the linear independent solutions

$$\begin{aligned} u_1(z) &= M(a, b, z), \\ u_2(z) &= z^{1-b}M(1+a-b, 2-b, z). \end{aligned}$$

So the general solution of (6.12) is

$$g(z) = AM(a, b, z) + Bz^{1-b}M(1+a-b, 2-b, z).$$

Using the method of variation of parameters, a particular solution of

$$a_0(z)g''(z) + a_1(z)g'(z) + a_2(z)g(z) = f(z)$$

is

$$g_p(z) = a(z)u_1(z) + b(z)u_2(z),$$

where

$$\begin{aligned} a'(z) &= -\frac{u_2(z)}{W(z)}\frac{f(z)}{a_0(z)}, \\ b'(z) &= \frac{u_1(z)}{W(z)}\frac{f(z)}{a_0(z)}, \\ W(z) &= u_1(z)u_2'(z) - u_2(z)u_1'(z). \end{aligned}$$

Using (6.2),

$$W(z) = (1-b)z^{-b}e^{z}.$$

Thus

$$\begin{aligned} a'(z) &= -\frac{z^{1-b}M(1+a-b,2-b,z)}{(1-b)z^{-b}e^{z}}\frac{f(z)}{z}, \\ b'(z) &= \frac{M(a,b,z)}{W(z)}\frac{f(z)}{z}. \end{aligned}$$

Applying the Kummer transformations [1, Equations 13.1.27 and 13.1.28] leads to

$$\begin{aligned} a'(z) &= -\frac{z^{1-b}M(1+a-b,2-b,z)}{(1-b)z^{-b}e^{z}}\frac{f(z)}{z} \\ &= -\frac{z^{1-b}e^{z}M(1-a,2-b,-z)}{(1-b)z^{-b}e^{z}}\frac{f(z)}{z} \\ &= -\frac{M(1-a,2-b,-z)}{(1-b)}f(z), \\ b'(z) &= \frac{M(a,b,z)}{(1-b)z^{-b}e^{z}}\frac{f(z)}{z} \\ &= \frac{e^{z}M(b-a,b,-z)}{(1-b)z^{-b}e^{z}}\frac{f(z)}{z} \\ &= \frac{z^{b-1}M(b-a,b,-z)}{(1-b)}f(z). \end{aligned}$$

Now we need to find an expression for $f(z)$. Recall that

$$\begin{aligned} x &= \mathcal{L}_r\phi(x) \\ &= \frac{1}{2}\sigma_D^2x^2\phi''(x) + \eta(\overline{D}-x)x\phi'(x) - r\phi(x) \\ &= x^{a+1}\left\{\frac{1}{2}\sigma^2xh''(x) + (\eta(\overline{D}-x)+\sigma^2a)h'(x) - \eta ah(x)\right\}. \end{aligned}$$

Substituting $x = \frac{\sigma^2 z}{2\eta}$,

$$\frac{\sigma^2 z}{2\eta} = \left(\frac{\sigma^2 z}{2\eta}\right)^{a+1}\left\{\eta\left(zg''(z) + g'(z)[b-z] - ag(z)\right)\right\}.$$

Thus

$$\begin{aligned} f(z) &= zg''(z) + (b-z)g'(z) - ag(z) \\ &= \frac{1}{\eta}\left(\frac{2\eta}{\sigma^2}\right)^{a+1} z^{-a-1}\frac{\sigma^2 z}{2\eta}. \end{aligned}$$

$$
\begin{aligned}
a'(z) &= -\frac{M(1-a,2-b,-z)}{(1-b)}\frac{1}{\eta}\left(\frac{2\eta}{\sigma^2}\right)^a z^{-a} \\
&= -\frac{1}{\eta(1-b)}\left(\frac{2\eta}{\sigma^2}\right)^a z^{-a}M(1-a,2-b,-z), \\
b'(z) &= \frac{z^{b-1}M(b-a,b,-z)}{(1-b)}\frac{1}{\eta}\left(\frac{2\eta}{\sigma^2}\right)^a z^{-a} \\
&= \frac{1}{\eta(1-b)}\left(\frac{2\eta}{\sigma^2}\right)^a z^{b-a-1}M(b-a,b,-z).
\end{aligned}
$$

Consider the following functions

$$
\begin{aligned}
f_1(z) &= z^{1-a}Y\left(1-a,2-b,-z\right), \\
f_2(z) &= z^{b-a}Y\left(b-a,b,-z\right),
\end{aligned}
$$

where

$$
Y(a,b,z) = \sum_{n=0}^{\infty}\frac{(a)_n}{(b)_n}\frac{z^n}{(a+n)n!}.
$$

By Lemma B.12,

$$
\begin{aligned}
\frac{\partial f_1(z)}{\partial z} &= \frac{2\eta}{\sigma^2}z^{-a}M\left(1-a,2-b,-z\right), \\
\frac{\partial f_2(z)}{\partial z} &= \frac{2\eta}{\sigma^2}z^{b-a-1}M\left(b-\theta,b,-z\right),
\end{aligned}
$$

and so we can choose

$$
\begin{aligned}
g_p(z) &= \frac{1}{\eta(1-b)}\left(\frac{2\eta}{\sigma^2}\right)^a z^{1-b}M(1+a-b,2-b,z)z^{b-a}Y(b-a,b,-z) \\
&\quad -\frac{1}{\eta(1-b)}\left(\frac{2\eta}{\sigma^2}\right)^a M(a,b,z)z^{1-a}Y(1-a,2-b,-z) \\
&= \frac{2}{\sigma^2(1-b)}\left(\frac{2\eta}{\sigma^2 z}\right)^{a-1} g_1(z),
\end{aligned}
$$

where

$$
g_1(z) = M(1+a-b,2-b,z)Y(b-a,b,-z) - M(a,b,z)Y(1-a,2-b,-z).
$$

Thus

$$\begin{aligned}\phi_p(x) &= x^a g_p\left(\frac{2\eta x}{\sigma^2}\right)\\ &= x^a \frac{2}{\sigma^2(1-b)} x^{1-a} g_1\left(\frac{2\eta x}{\sigma^2}\right)\\ &= \frac{2\Phi(x)}{\sigma^2(1-b)},\end{aligned}$$

where

$$\begin{aligned}\Phi(x) &= x g_1\left(\frac{2\eta x}{\sigma^2}\right)\\ &= xM\left(1+a-b, 2-b, \frac{2\eta x}{\sigma^2}\right) Y\left(b-a, b, \frac{-2\eta x}{\sigma^2}\right)\\ &-xM\left(a, b, \frac{2\eta x}{\sigma^2}\right) Y\left(1-a, 2-b, \frac{-2\eta x}{\sigma^2}\right).\end{aligned}$$

The general solution is

$$\phi(x) = A_1\psi_1(x) + A_2\psi_2(x) + \frac{2\Phi(x)}{\sigma^2(1-b)}.$$

In order that ϕ be bounded near 0, $A_2 = 0$ for $b > 1+a$. Thus the general solution is

$$\phi(x) = A\psi_1(x) + \frac{2\Phi(x)}{\sigma^2(1-b)}.$$

Now consider $x > S$,

$$\mathcal{L}_r\phi(x) = S.$$

The general solution of $\mathcal{L}_r\phi = 0$ is

$$\phi(x) = B_1\psi_1(x) + B_2\psi_2(x).$$

A particular solution is

$$\phi_p(x) = \frac{-S}{r}.$$

We require $\phi(x)$ bounded as $x \to \infty$, but both $\psi_1(x)$ and $\psi_2(x)$ are unbounded (Lemmas B.20 and B.21). Fortunately, $\psi_4(x)$ is bounded (Lemma B.22) and using

(6.1) we can choose B_1 and B_2 so that

$$B_1\psi_1(x) + B_2\psi_2(x) = B\psi_4(x).$$

Thus the general solution is

$$\phi(x) = B\psi_4(x) - \frac{S}{r}.$$

For ϕ smooth, $\phi(S^-) = \phi(S^+)$ and $\phi'(S^-) = \phi'(S^+)$

$$\begin{aligned} A\psi_1(S) + \frac{2\Phi(S)}{\sigma^2(1-b)} &= \frac{-S}{r} + B\psi_4(S), \\ A\psi_1'(S) + \frac{2\Phi'(S)}{\sigma^2(1-b)} &= B\psi_4'(S). \end{aligned}$$

Re-arranging leads to

$$\begin{aligned} A &= \frac{\frac{2\Phi\ (S)\psi_4(S)}{\sigma^2(1-b)} - \frac{2\Phi(S)\psi_4(S)}{\sigma^2(1-b)} - \frac{S\psi_4(S)}{r}}{W(\psi_1(S), \psi_4(S))}, \\ B &= \frac{\frac{2\Phi\ (S)\psi_1(S)}{\sigma^2(1-b)} - \frac{2\Phi(S)\psi_1(S)}{\sigma^2(1-b)} - \frac{S\psi_1(S)}{r}}{W(\psi_1(S), \psi_4(S))}. \end{aligned}$$

Using Lemma B.14,

$$\begin{aligned} \Phi'(S)\psi_4(S) - \Phi(S)\psi_4'(S) &= S^{b-a}Y\left(b-a, b, \frac{-2\eta S}{\sigma^2}\right) W(\psi_2(S), \psi_4(S)) \\ &\quad - S^{1-a}Y\left(1-a, 2-b, \frac{-2\eta S}{\sigma^2}\right) W(\psi_1(S), \psi_4(S)), \\ \Phi'(S)\psi_1(S) - \Phi(S)\psi_1'(S) &= S^{b-a}Y\left(b-a, b, \frac{-2\eta S}{\sigma^2}\right) W(\psi_2(S), \psi_1(S)) \\ &\quad - S^{1-a}Y\left(1-a, 2-b, \frac{-2\eta S}{\sigma^2}\right) W(\psi_1(S), \psi_1(S)). \end{aligned}$$

Applying Lemma B.13,

$$\begin{aligned} W(\psi_4(S), \Phi(S)) &= S^{b-a} Y\left(b-a, b, \frac{-2\eta S}{\sigma^2}\right) S^{2a-b} \frac{-\Gamma(2-b)}{\Gamma(1+a-b)} \exp\left(\frac{2\eta S}{\sigma^2}\right) \\ &\quad - S^{1-a} Y\left(1-a, 2-b, \frac{-2\eta S}{\sigma^2}\right) S^{2a-b} \frac{-\Gamma(b)}{\Gamma(a)} \exp\left(\frac{2\eta S}{\sigma^2}\right)\left(\frac{2\eta}{\sigma^2}\right)^{1-b}, \\ W(\psi_1(S), \Phi(S)) &= S^{b-a} Y\left(b-a, b, \frac{-2\eta S}{\sigma^2}\right) S^{2a-b}\left\{-(1-b)\exp\left(\frac{2\eta S}{\sigma^2}\right)\right\}, \\ W(\psi_1(S), \psi_4(S)) &= S^{2a-b} \frac{-\Gamma(b)}{\Gamma(a)} \exp\left(\frac{2\eta S}{\sigma^2}\right)\left(\frac{2\eta S}{\sigma^2}\right)^{1-b}. \end{aligned}$$

Thus

$$\begin{aligned} A &= \frac{\Gamma(a)}{\Gamma(b)}\left(\frac{2\eta}{\sigma^2}\right)^{b-1}\left\{\frac{S^{1+b-2a}\psi_4'(S)}{r\exp\left(\frac{2\eta S}{\sigma^2}\right)} - \frac{2\Gamma(1-b)S^{b-a}}{\sigma^2\Gamma(1+a-b)} Y\left(b-a, b, \frac{-2\eta S}{\sigma^2}\right)\right\} \\ &\quad + \frac{2S^{1-a} Y\left(1-a, 2-b, \frac{-2\eta S}{\sigma^2}\right)}{\sigma^2(1-b)}, \\ B &= \frac{\Gamma(a)}{\Gamma(b)}\left(\frac{2\eta}{\sigma^2}\right)^{b-1}\left\{\frac{S^{1+b-2a}\psi_1'(S)}{r\exp\left(\frac{2\eta S}{\sigma^2}\right)} - \frac{2S^{b-a}}{\sigma^2} Y\left(b-a, b, \frac{-2\eta S}{\sigma^2}\right)\right\}. \end{aligned}$$

We note that $|x\phi'(x)| \leq C$ for all $x \geq 0$. For $x \in [0, S]$,

$$\begin{aligned} |x\phi'(x)| &= \left|\frac{2x\Phi'(x)}{\sigma^2(1-b)} + aAx^a M\left(a+1, b, \frac{2\eta x}{\sigma^2}\right)\right| \\ &< \infty. \end{aligned}$$

For $x \in [S, \infty)$, $|x\phi'(x)| < \infty$, since $|S\phi'(S)| < \infty$ and using (6.6),

$$\begin{aligned} \lim_{x\to\infty} |x\phi'(x)| &= \lim_{x\to\infty} ax^a U\left(a+1, b, \frac{2\eta x}{\sigma^2}\right) \\ &= \lim_{x\to\infty} ax^a \left(\frac{2\eta x}{\sigma^2}\right)^{-a-1}\left[1 + O\left(\left|\frac{2\eta x}{\sigma^2}\right|^{-1}\right)\right] = 0. \end{aligned}$$

Using Lemmas B.18 and B.19, we can show that $\Phi'(x)$, $M\left(a+1, b, \frac{2\eta x}{\sigma^2}\right)$ and $U\left(a+1, b, \frac{2\eta x}{\sigma^2}\right)$ are convergent for all $x < \infty$.

We also note that ϕ is twice differentiable. We only need to show that their second derivatives match at S,

$$\phi''(S-) = \phi''(S+).$$

Re-arranging the PDE

$$\phi''(x) = \frac{\min(S,x) + r\phi(x) - \eta(\overline{D} - x)x\phi'(x)}{\frac{1}{2}x^2\sigma^2}, \quad x \neq S.$$

Since

$$\begin{aligned} \phi(S+) &= \phi(S-), \\ \phi'(S+) &= \phi'(S-), \end{aligned}$$

we must have

$$\phi''(S+) = \phi''(S-).$$

Finally, we note that ϕ is unique. Suppose that ϕ_1 and ϕ_2 are two solutions. Let

$$\overline{\phi} = \phi_1 - \phi_2,$$

then

$$\mathcal{L}_r\overline{\phi} = 0.$$

The general solution is

$$\overline{\phi}(x) = Ax^\theta M\left(\theta, b, \frac{2\eta x}{\sigma^2}\right) + Bx^{\theta+1-b}M\left(\theta+1-b, 2-b, \frac{2\eta x}{\sigma^2}\right).$$

Since $\overline{\phi}$ bounded, we must have $B = 0$. Since $\overline{\phi}(0) = 0$, we must have $A = 0$. Therefore $\overline{\phi} = 0$.

□

6.3.4 The discount factors $E\left[e^{-r\tau}\right]$ and $E\left[e^{-(r+\alpha)\tau}\right]$

Lemma 6.4 provides expressions for the discount factors $E[e^{-r\tau}]$ and $E[e^{-(r+\alpha)\tau}]$,

$$\begin{aligned} E\left[e^{-r\tau}\right] &= \frac{\psi_1(D)}{\psi_1(y)}, \\ E\left[e^{-(r+\alpha)\tau}\right] &= \frac{\psi_3(D)}{\psi_3(y)}. \end{aligned}$$

Figure 6.3 shows the discount factors for selected logistic versions of Example 5.1a. As expected, the discount functions converge to that of the GBM model as $\overline{D} \to \infty$.

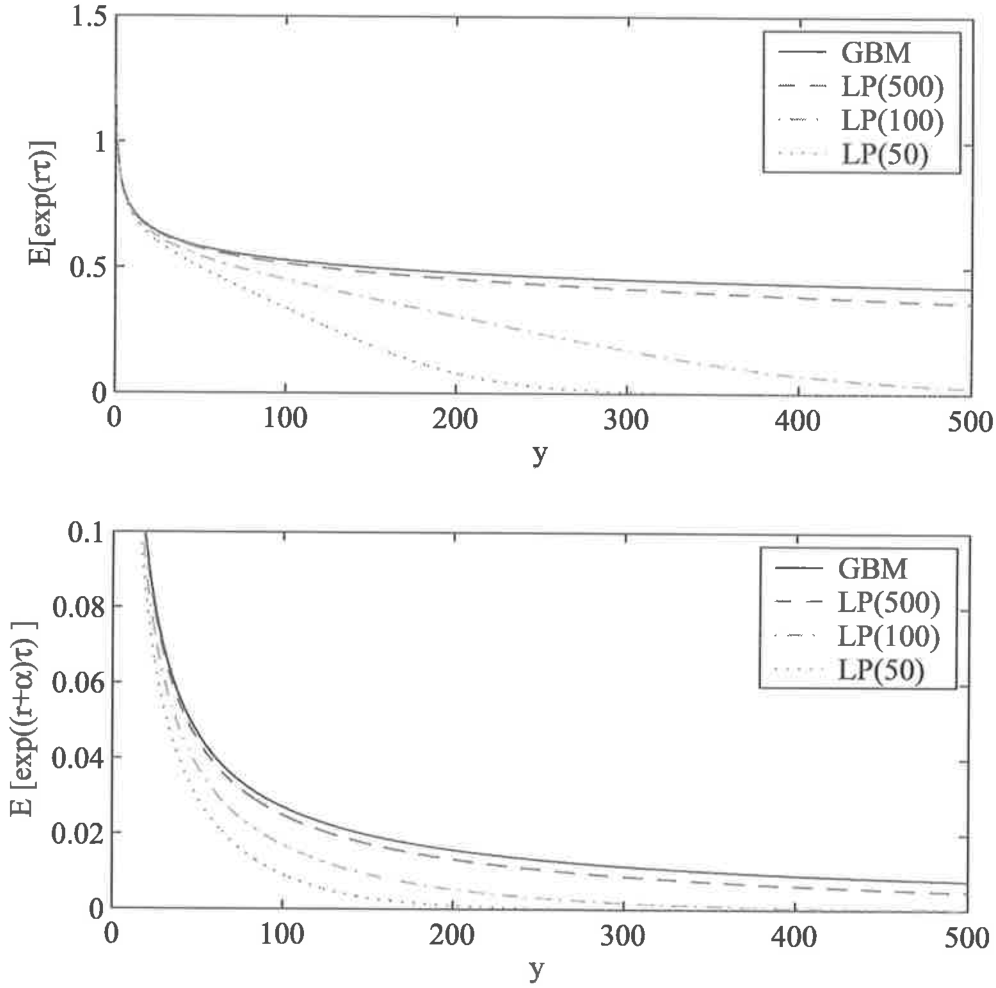


Figure 6.3: Discount factors for logistic versions of Example 5.1a

Lemma 6.4.

$$E\left[e^{-\lambda\tau}\right] = \left(\frac{D}{y}\right)^{a} \frac{M\left(a, b, \frac{2\eta D}{\sigma^2}\right)}{M\left(a, b, \frac{2\eta y}{\sigma^2}\right)},$$

where

$$\begin{aligned} a &= \frac{-\left[\eta\overline{D} - \frac{\sigma^2}{2}\right] + \sqrt{\left(\eta\overline{D} - \frac{\sigma^2}{2}\right)^2 + 2\lambda\sigma^2}}{\sigma^2}, \\ b &= \frac{2\overline{D}\eta}{\sigma^2} + 2a. \end{aligned}$$

Proof. We seek a unique, bounded solution of

$$\frac{1}{2}\sigma^2\phi'' + \eta(\overline{D} - x)x\phi'(x) - \lambda\phi(x) = 0, \tag{6.13}$$

with $|x\phi'(x)|$ bounded. By Lemma 6.9, the general solution is

$$\phi(x) = Ax^a M\left(a, b, \frac{2\eta x}{\sigma^2}\right) + Bx^{a+1-b} M\left(a+1-b, 2-b, \frac{2\eta x}{\sigma^2}\right),$$

where a is the positive root of

$$\frac{1}{2}\sigma^2 a(a-1) + \eta\overline{D}a - \lambda = 0,$$

and $b = \frac{2\overline{D}\eta}{\sigma^2} + 2a$. For $\phi(0) = 0$, the general solution is simply

$$\phi(x) = Ax^a M\left(a, b, \frac{2\eta x}{\sigma^2}\right).$$

By Lemma B.7,

$$\phi'(x) = Aax^{a-1} M\left(a, b, \frac{2\eta x}{\sigma^2}\right).$$

For $x \in [0, y]$,

$$|x\phi'(x)| \leq Aa\left|y^a M\left(a, b, \frac{2\eta y}{\sigma^2}\right)\right|.$$

Thus $|x\phi'(x)|$ is bounded on $[0, y]$. Applying Theorem 5.2,

$$\begin{aligned} E\left[e^{-\lambda\tau}\right] &= \frac{\phi(D)}{\phi(y)} \\ &= \frac{AD^a M\left(a, b, \frac{2\eta D}{\sigma^2}\right)}{Ay^a M\left(a, b, \frac{2\eta y}{\sigma^2}\right)} \\ &= \left(\frac{D}{y}\right)^a \frac{M\left(a, b, \frac{2\eta D}{\sigma^2}\right)}{M\left(a, b, \frac{2\eta y}{\sigma^2}\right)}. \end{aligned} \tag{6.14}$$

□

6.3.5 The Optimal Trigger

In Section 5.3.5 we divided the investment function $F(D, y)$ into three regions: $[0, S_0)$, $[S_0, S_1)$ and $[S_1, \infty)$, and then established four results for the GBM model:

1. $F'(D, y) > 0$ on $[0, S_0)$ (Lemma 5.10),
2. $F'(D, y) = 0 \Rightarrow F''(D, y) < 0$ on $[S_0, S_1)$ (Lemma 5.12),
3. $F''(D, y) < 0$ on $[S_1, \infty)$ (Lemma 5.14),
4. $\lim_{y\to\infty} F'(D, y) < 0$ (Lemma 5.15).

These results implied that a unique optimal trigger exists and that it may reside in either the second or third region. We would like to establish similar results for the logistic model. It is not difficult to establish that Result 1 also holds for the logistic model (see Lemma 6.5). We could neither find any counter examples to refute Results 2 and 3 nor establish these results for the logistic model. For $I >> \beta$, $F'(D, y)$ may converge to a positive value (Lemma 6.6). In this case there is no optimal trigger and since it is always better to wait the investment will be postponed indefinitely.

Lemma 6.5.

$$F'(D, y) = \beta \frac{\psi_1(D)}{\psi_1(y)^2} f(y) + \frac{I\psi_3(D)\psi_3'(y)}{\psi_3(y)^2},$$

where

$$f(y) = \begin{cases} 0, & 0 \le y \le S_0; \\ \frac{S_0}{r}\psi_1'(y) - \left(\frac{2}{\sigma^2}\psi_8(y) + \frac{\Gamma(b)}{\Gamma(a)}\left(\frac{2\eta}{\sigma^2}\right)^{1-b} B_0\right) y^{2a-b} \exp\left(\frac{2\eta y}{\sigma^2}\right) y^{2a-b}, & S_0 \le y \le S_1; \\ \frac{S_0 - S_1}{r}\psi_1'(y) - \frac{\Gamma(b)}{\Gamma(a)}\left(\frac{2\eta}{\sigma^2}\right)^{1-b} (B_0 - B_1) y^{2a-b} \exp\left(\frac{2\eta y}{\sigma^2}\right), & S_1 \le y < \infty. \end{cases}$$

Proof. For $y \in [0, S_0]$, the investment value is given by

$$\begin{aligned} F(D,y) &= \beta\left(A_0\psi_1(y) + \frac{2\Phi(y)}{\sigma^2(b-1)}\right)\frac{\psi_1(D)}{\psi_1(y)} - \beta\phi_0(D) + \beta C \\ &\quad -\beta\left(A_1\psi_a(y) + \frac{2\Phi(y)}{\sigma^2(b-1)}\right)\frac{\psi_1(D)}{\psi_1(y)} + \beta\phi_1(D) - I\frac{\psi_3(D)}{\psi_3(y)} \\ &= \beta(A_0\psi_1(D) - \phi_0(D) + C - A_1\psi_1(D) + \phi_1(D)) - I\frac{\psi_3(D)}{\psi_3(y)} \\ &= C_1 - I\frac{\psi_3(D)}{\psi_3(y)}, \end{aligned}$$

where $C_1 = \beta[A_0\psi_1(D) - \phi_0(D) + C - A_1\psi_1(D) + \phi_1(D)]$. Thus

$$\begin{aligned} \frac{\partial F(D,y)}{\partial y} &= -I\psi_3(D)(-(\psi_3(y))^{-2})\psi_3'(y) \\ &= \frac{I\psi_3(D)\psi_3'(y)}{\psi_3(y)^2}. \end{aligned}$$

For $y \in [S_0, S_1]$, the investment value is given by

$$\begin{aligned} F(D,y) &= \beta\left(-\frac{S_0}{r} + B_0\psi_4(y)\right)\frac{\psi_1(D)}{\psi_1(y)} - \beta\phi_0(y) + \beta C \\ &\quad -\beta\left(A_1\psi_1(y) + \frac{2\Phi(y)}{\sigma(b-1)}\right)\frac{\psi_1(D)}{\psi_1(y)} + \beta\phi_1(y) - I\frac{\psi_3(D)}{\psi_3(y)} \\ &= \beta(A_0\psi_1(D) - \phi_0(y) + C + \phi_1(y)) \\ &\quad +\beta\left\{\frac{2\Phi(y)}{\sigma^2(b-1)} + \frac{S_1}{r} - B_1\psi_4(y)\right\}\frac{\psi_1(D)}{\psi_1(y)} - I\frac{\psi_3(D)}{\psi_3(y)}. \end{aligned}$$

Applying Lemma B.13,

$$\begin{aligned} W(\psi_1(y), \Phi(y),) &= -(1-b)y^{2a-b}\exp\left(\frac{2\eta y}{\sigma^2}\right)\psi_8(y), \\ W(\psi_1(y), \psi_4(y)) &= -\frac{\Gamma(b)}{\Gamma(a)}\left(\frac{2\eta S}{\sigma^2}\right)^{1-b}. \end{aligned}$$

Thus

$$\begin{aligned} \frac{\partial F(D,y)}{\partial y} &= \beta\psi_1(D)\frac{-S_0}{r}(-(\psi 1(y))^{-2})\psi_1'(y) \\ &\quad +\beta\psi_1(D)B_0\frac{-\Gamma(b)}{\Gamma(a)}\left(\frac{2\eta S}{\sigma^2}\right)^{1-b} y^{2a-b}\exp\left(\frac{2\eta S}{\sigma^2}\right)(\psi_1(y))^{-2} \\ &\quad -\beta\psi_1(D)\frac{2}{\sigma^2(b-1)}(-(1-b))y^{2a-b}\exp\left(\frac{2\eta y}{\sigma^2}\right)\psi_8(y)(\psi_1(y))^{-2} \\ &\quad -I\psi_3(D)(-(\psi_3(y))^{-2})\psi_3'(y) \\ &= \beta\frac{\psi_1(D)}{\psi_1(y)^2}f_1(y) + \frac{I\psi_3(D)\psi_3'(y)}{\psi_3(y)^2}, \end{aligned}$$

where

$$f_1(y) = \frac{S_0}{r}\psi_1'(y) - \left\{\frac{2}{\sigma^2}\psi_8(y) - \frac{\Gamma(b)}{\Gamma(a)}\left(\frac{2\eta}{\sigma^2}\right)^{1-b} B_0\right\} y^{2a-b}\exp\left(\frac{2\eta S}{\sigma^2}\right).$$

For $y \in [S_1, \infty]$, the investment value is given by

$$\begin{aligned}
F(D,y) &= \beta\left(-\frac{S_0}{r} + B_0\psi_4(y)\right)\frac{\psi_1(D)}{\psi_1(y)} - \beta\phi_0(y) + \beta C \\
&\quad -\beta\left(-\frac{S_1}{r} + B_1\psi_4(y)\right)\frac{\psi_1(D)}{\psi_1(y)} + \beta\phi_1(y) - I\frac{\psi_3(D)}{\psi_3(y)} \\
&= \beta(-\phi_0(y) + C + \phi_1(y)) \\
&\quad +\beta\left\{\frac{S_1 - S_0}{r} - (B_1 - B_0)\psi_4(y)\right\}\frac{\psi_1(D)}{\psi_1(y)} - I\frac{\psi_3(D)}{\psi_3(y)}.
\end{aligned}$$

Thus

$$\begin{aligned}
\frac{\partial F(D,y)}{\partial y} &= \beta\psi_1(D)\frac{S_1 - S_0}{r}(-(\psi 1(y))^{-2})\psi_1'(y) \\
&\quad +\beta\psi_1(D)(B_0 - B_1)\frac{-\Gamma(b)}{\Gamma(a)}\left(\frac{2\eta S}{\sigma^2}\right)^{1-b} y^{2a-b}\exp\left(\frac{2\eta S}{\sigma^2}\right)(\psi_1(y))^{-2} \\
&\quad -I\psi_3(D)(-(\psi_3(y))^{-2})\psi_3'(y) \\
&= \beta\frac{\psi_1(D)}{\psi_1(y)^2}f_2(y) + \frac{I\psi_3(D)\psi_3'(y)}{\psi_3(y)^2},
\end{aligned}$$

where

$$f_2(y) = \frac{S_0 - S_1}{r}\psi_1'(y) - \frac{\Gamma(b)}{\Gamma(a)}\left(\frac{2\eta}{\sigma^2}\right)^{1-b}(B_0 - B_1)y^{2a-b}\exp\left(\frac{2\eta S}{\sigma^2}\right).$$

□

Lemma 6.6.

$$\lim_{y\to\infty}\frac{(\psi_1(y))^2F'(D,y)}{\beta\psi_1(D)\psi_1'(y)} = \frac{S_0 - S_1}{r} - (B_0 - B_1)\left(\frac{2\eta}{\sigma^2}\right)^{-a} + \frac{I\psi_3(D)}{\beta\psi_1(D)}\left(\frac{2\eta}{\sigma^2}\right)^{a_1 - a}\frac{\Gamma(b)\Gamma(a_1)}{\Gamma(b_1)\Gamma(a)}.$$

Proof. Let

$$\begin{aligned}
g_1(y) &= \frac{S_0 - S_1}{r}, \\
g_2(y) &= -\frac{\Gamma(b)}{\Gamma(a)}\left(\frac{2\eta}{\sigma^2}\right)^{1-b}(B_0 - B_1)\frac{y^{2a-b}\exp\left(\frac{2\eta S}{\sigma^2}\right)}{\psi_1'(y)}, \\
g_3(y) &= \frac{I\psi_3(D)}{\beta\psi_1(D)}\frac{\psi_3'(y)(\psi_1(y))^2}{(\psi_3(y))^2\psi_1'(y)},
\end{aligned}$$

then

$$F'(D,y) = \frac{\beta\psi_1(D)\psi_1'(y)}{(\psi_1(y))^2}\{g_1(y)+g_2(y)+g_3(y)\}.$$

Using (6.5),

$$\begin{aligned}
\lim_{y\to\infty} g_2(y) &= \lim_{y\to\infty} \frac{-\frac{\Gamma(b)}{\Gamma(a)}\left(\frac{2\eta}{\sigma^2}\right)^{1-b}(B_0-B_1)y^{2a-b}\exp\left(\frac{2\eta S}{\sigma^2}\right)}{ay^{a-1}\frac{\Gamma(b)}{\Gamma(a+1)}\exp\left(\frac{2\eta y}{\sigma^2}\right)\left(\frac{2\eta y}{\sigma^2}\right)^{a+1-b}[1+O(|\frac{2\eta y}{\sigma^2}|^{-1})]} \\
&= -(B_0-B_1)\left(\frac{2\eta}{\sigma^2}\right)^{-a}.
\end{aligned}$$

Using (6.6) and observing that $2a-b = 2a_1-b_1 = 2\eta\overline{D}$,

$$\begin{aligned}
\lim_{y\to\infty} g_3(y) &= \lim_{y\to\infty} \frac{I\psi_3(D)}{\beta\psi_1(D)}\left(\frac{y^a\frac{\Gamma(b)}{\Gamma(a)}\exp\left(\frac{2\eta y}{\sigma^2}\right)\left(\frac{2\eta y}{\sigma^2}\right)^{a-b}[1+O(|\frac{2\eta y}{\sigma^2}|^{-1})]}{y^{a_1}\frac{\Gamma(b_1)}{\Gamma(a_1)}\exp\left(\frac{2\eta y}{\sigma^2}\right)\left(\frac{2\eta y}{\sigma^2}\right)^{a_1-b_1}[1+O(|\frac{2\eta y}{\sigma^2}|^{-1})]}\right)^2 \\
&\quad\times\frac{a_1y^{a_1-1}\frac{\Gamma(b_1)}{\Gamma(a_1+1)}\exp\left(\frac{2\eta y}{\sigma^2}\right)\left(\frac{2\eta y}{\sigma^2}\right)^{a_1+1-b_1}[1+O(|\frac{2\eta y}{\sigma^2}|^{-1})]}{ay^{a-1}\frac{\Gamma(b)}{\Gamma(a+1)}\exp\left(\frac{2\eta y}{\sigma^2}\right)\left(\frac{2\eta y}{\sigma^2}\right)^{a+1-b}[1+O(|\frac{2\eta y}{\sigma^2}|^{-1})]} \\
&= \frac{I\psi_3(D)}{\beta\psi_1(D)}\left(\frac{2\eta}{\sigma^2}\right)^{a_1-a}\frac{\Gamma(b)\Gamma(a_1)}{\Gamma(b_1)\Gamma(a)}.
\end{aligned}$$

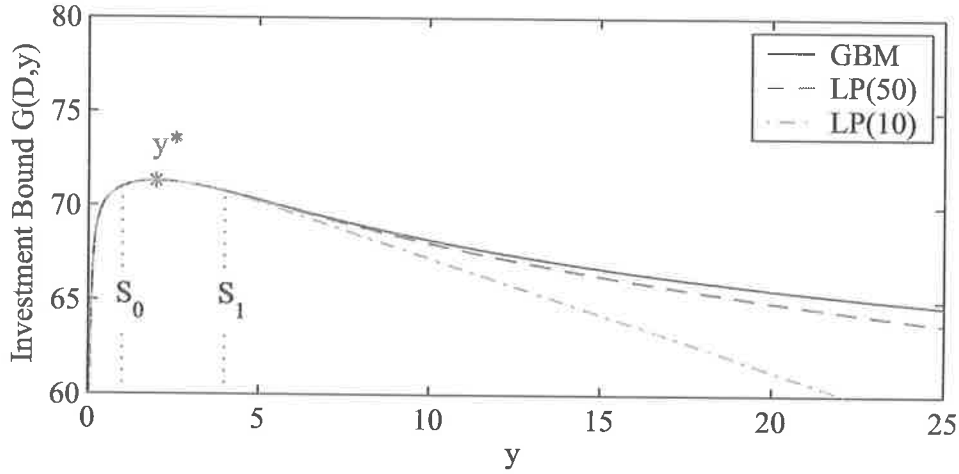
□

6.3.6 Numerical Examples

In this section we shall use Examples 5.1a and 5.1b to compare the logistic and GBM models. As discussed earlier, we define a logistic variation of a GBM by choosing $\overline{D}$ arbitrarily and setting $\eta = \frac{\nu}{\overline{D}}$. Figures 6.4 and 6.5 show some investment bounds for Example 5.1a and 5.1b respectively. We observe that the investment bounds and derivatives are close near the optimal trigger even though they diverge in other regions. We expect demand processes with larger values of $\overline{D}$ to yield higher investment values, as

$$F(D,y) = E\left[\int_0^\tau \beta\min(D(t),S_0)e^{-rt}dt + \int_\tau^\infty \beta\min(D(t),S_1)e^{-rt}dt - Ie^{-(r+\alpha)\tau}\right].$$

So it is surprising to note that their investment bounds are in fact lower (see Tables 6.1 and 6.2). Since the investment bound $G(D,y)$ assumes the same upper



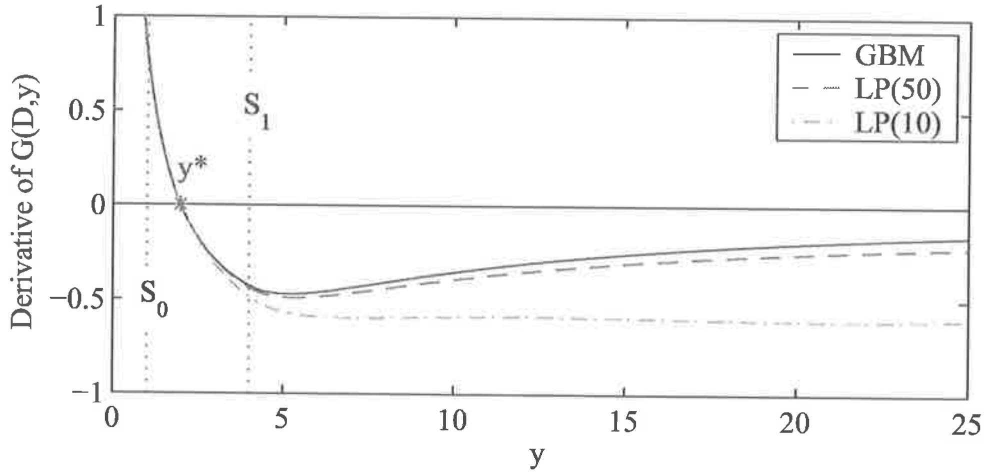


Figure 6.4: Investment bounds for logistic variations of Example 5.1a

bound $\frac{S_1}{r}$ for C for all $\overline{D}$, the investment bound does not provide enough information to compare the investment values for the GBM and logistic models. We used simulation techniques to estimate the investment values and stopping time distributions. The geometric Brownian motions and logistic processes were simulated using Euler's method [37] for N=10000 runs, and the same seed was used to instantiate each process. Tables 6.3 and 6.4 show the 95% confidence intervals for the invest-

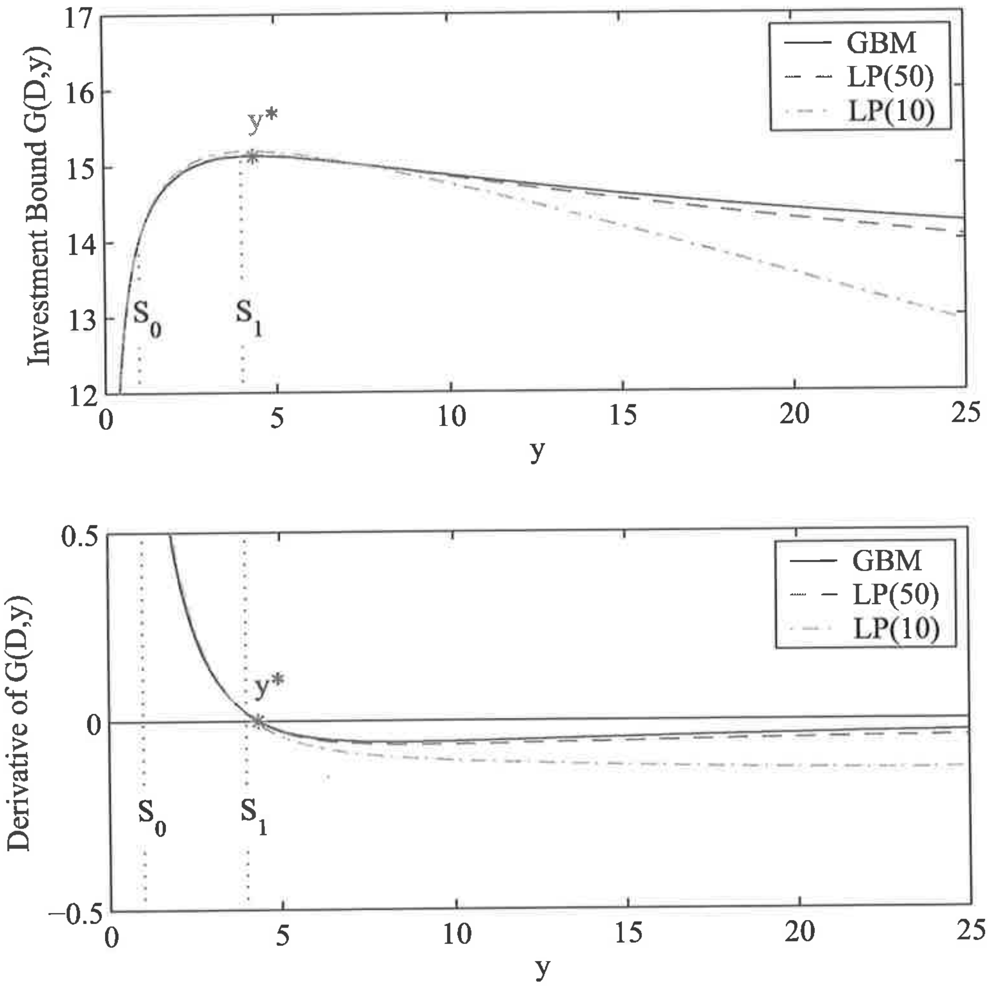


Figure 6.5: Investment bounds for logistic variations of Example 5.1b

ment values and stopping time distributions for Examples 5.1a and 5.1b. From (2.4), we know that the stopping times are almost surely finite in the GBM model when $\nu_D > \frac{\sigma_D^2}{2}$. We observe that larger values of $\overline{D}$ yield earlier stopping times and larger investment values. These results provide some insight into what happens when a GBM is used to approximate a logistic process. In each example, the GBM model predicts inflated investment values and recommends a different trigger level (e.g. a higher trigger in Example 5.1a and a lower trigger in Example 5.1b). Following this strategy will yield suboptimal investment values, but Figures 6.4 and 6.5 suggest that the reduction in value will be small. So the GBM is a good approximation in these examples.

$\overline{D}$	$\eta\overline{D}$	σ_D	r	D	S_0	S_1	I	β	y^*	$G(D,y^*)$
10	0.75	0.95	0.05	1	0	4	1	0.9	1.9935	71.3316
100	0.75	0.95	0.05	1	0	4	1	0.9	1.9831	71.3128
1000	0.75	0.95	0.05	1	0	4	1	0.9	1.9821	71.3109
10000	0.75	0.95	0.05	1	0	4	1	0.9	1.9820	71.3107
∞	0.75	0.95	0.05	1	0	4	1	0.9	1.9819	71.3107

Table 6.1: Logistic variations of Example 5.1a.

$\overline{D}$	$\eta\overline{D}$	σ_D	r	D	S_0	S_1	I	β	y^*	$G(D,y^*)$
10	0.75	0.95	0.05	1	0	4	2	0.2	4.2417	15.1966
100	0.75	0.95	0.05	1	0	4	2	0.2	4.3431	15.1384
1000	0.75	0.95	0.05	1	0	4	2	0.2	4.3533	15.1326
10000	0.75	0.95	0.05	1	0	4	2	0.2	4.3543	15.1321
∞	0.75	0.95	0.05	1	0	4	2	0.2	4.3544	15.1320

Table 6.2: Logistic variations of Example 5.1b

$\overline{D}$	D^*	$P(\tau < 1000)$	$E[\tau \mid \tau < 1000]$	$G(D,y^*)$	$F(D,y^*)$
10	1.9935	1	2.99287 ± 0.125587	71.3316	38.00653 ± 0.228603
100	1.9831	1	2.608529 ± 0.113257	71.3128	53.485399 ± 0.260235
1000	1.9821	1	2.576671 ± 0.112577	71.3109	56.190303 ± 0.261484
10000	1.9820	1	2.574076 ± 0.11255	71.3107	56.573874 ± 0.261743
∞	1.9819	1	2.573749 ± 0.11255	71.3107	56.638977 ± 0.261877

Table 6.3: Simulation results for logistic variations Example 5.1a

$\overline{D}$	D^*	$P(\tau < 1000)$	$E[\tau\|\tau < 1000]$	$G(D, y^*)$	$F(D, y^*)$
10	4.2417	1	6.523659 ± 0.183835	15.1966	7.791029 ± 0.050801
100	4.3431	1	5.337618 ± 0.153946	15.1384	11.176755 ± 0.05783
1000	4.3533	1	5.235404 ± 0.151486	15.1326	11.772467 ± 0.058107
10000	4.3543	1	5.225129 ± 0.151331	15.1321	11.85725 ± 0.058165
∞	4.3544	1	5.224642 ± 0.151327	15.132	11.871617 ± 0.058195

Table 6.4: Simulation results for logistic variations Example 5.1b

In Chapter 5 we derived a separate formula for the special case $\nu_D = r$ and found that the GBM model supported the special case $S_0 = 0$. Neither of these special cases are supported in the logistic case. However, we can find estimates for these cases by taking limits in the logistic model. Recall that Example 5.2c exhibits both properties (i.e. $\nu_D = r$ and $S_0 = 0$). Table 6.5 shows the results for $r = \nu_D - 10^{-7}$ and $S_0 = 10^{-7}$. We note that the GBM (i.e. $\overline{D} = \infty$) results are correct to four decimal places, and expect that the logistic results are also close.

$\overline{D}$	$\eta\overline{D}$	σ_D	r	D	S_0	S_1	I	β	y^*	$G(D, y^*)$
10	0.04	0.2	$0.04 - 10^{-7}$	1	10^{-7}	4	5	0.2	1.5110	17.8389
100	0.04	0.2	$0.04 - 10^{-7}$	1	10^{-7}	4	5	0.2	1.5246	17.8540
1000	0.04	0.2	$0.04 - 10^{-7}$	1	10^{-7}	4	5	0.2	1.5260	17.8556
10000	0.04	0.2	$0.04 - 10^{-7}$	1	10^{-7}	4	5	0.2	1.5262	17.8557
∞	0.04	0.2	$0.04 - 10^{-7}$	1	10^{-7}	4	5	0.2	1.5262	17.8557
∞	0.04	0.2	0.04	1	0	4	5	0.2	1.5262	17.8557

Table 6.5: Logistic variations of Example 5.2c

In Chapter 5 we used the fact that $F'(D, y)$ converged to a negative number to prove that the optimal trigger exists in the GBM model. In the logistic model, however, $F'(D, y)$ may converge to a positive number. Figure 6.6 shows an example with this property: a logistic variation of Example 5.1a with $\overline{D} = 15$ and I =100, 500, 1000, and 2000. Further details are provided in Table 6.6. The optimal trigger

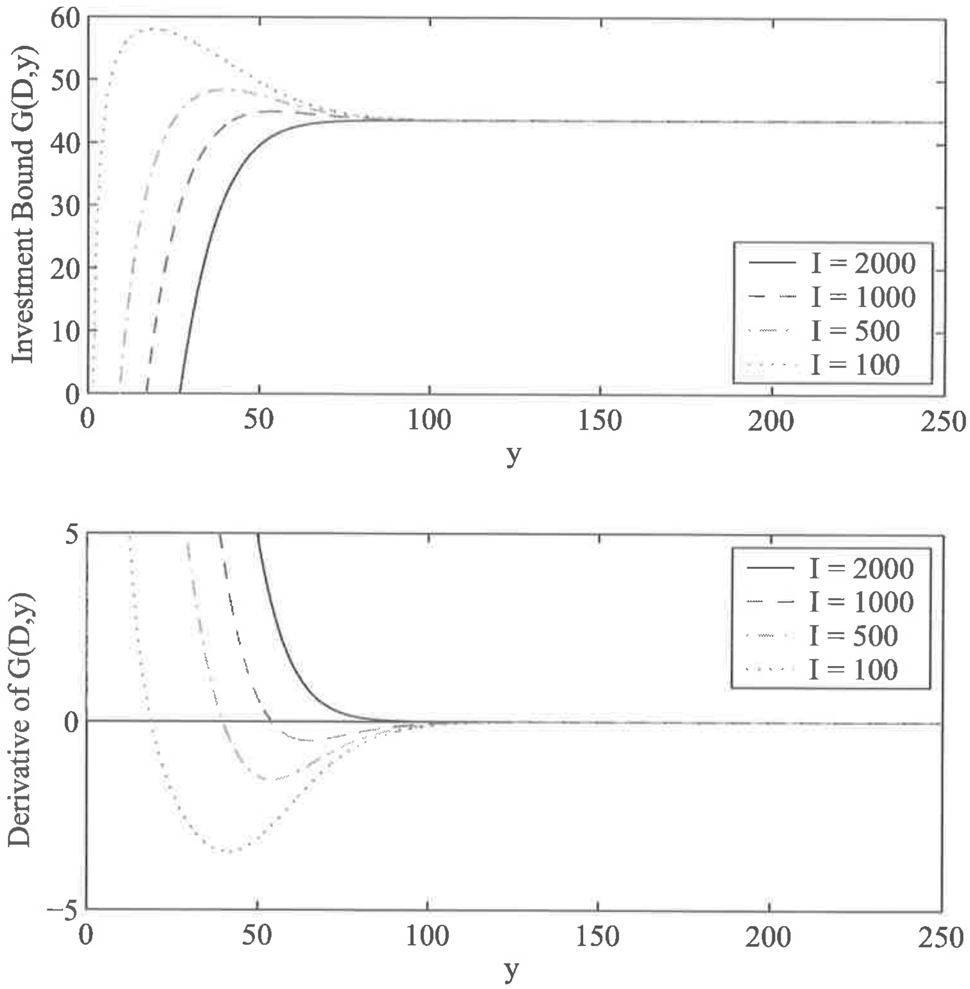


Figure 6.6: A logistic variation of Example 5.1a with large I

exists for low values of $\overline{D}$, but there is no optimal trigger when $I = 2000$. As discussed previously in Chapter 3, when there is no optimal trigger it is always better to wait and so the investment is never made. Yet the graph suggests that waiting yields diminishing returns. Hence, it may be pragmatic to choose a trigger $\widehat{y}$ with the property that $F'(D, \widehat{y})$ is less than some threshold ϵ.

$\overline{D}$	$\eta\overline{D}$	σ_D	r	D	S_0	S_1	I	β	y^*	$G(D,y^*)$
15	0.75	0.95	0.05	1	1	4	100	0.9	18.7835	57.9868
15	0.75	0.95	0.05	1	1	4	500	0.9	39.8442	48.4290
15	0.75	0.95	0.05	1	1	4	1000	0.9	53.9448	44.9741
15	0.75	0.95	0.05	1	1	4	2000	0.9	∞	0

Table 6.6: A logistic variation of Example 5.1a with large I

6.4 Conclusion

In this chapter we found an analytical solution for an increasing link capacity model with demand saturation. In the GBM Model, we obtained an explicit expression for $C = E\left[\int_0^\infty \min(D(s), S)e^{-rs}ds\right]$ and could therefore calculate the investment value $F(D, y)$. In the logistic model, we could not find an explicit expression for the C and so we calculated the investment bound $G(D, y)$. When greater precision was required, we used simulation techniques to estimate C and thereby estimate $F(D, y)$. These simulations were also used to estimate the stopping times. For low demand saturation levels, we found that the estimated investment values were significantly (nearly 50%) less than the investment bounds. While the logistic model does not explicitly support two special cases ($S_0 = 0$ and $\eta\overline{D} = r$) which are supported by the GBM model, we were able to estimate these values by taking limits. One surprising property of the logistic model is that it may not have an optimal trigger, meaning that it is always better to wait. However, waiting offers diminishing returns and so it may be pragmatic to invest when the demand reaches the lowest threshold at which the investment value is within some neighbourhood of the optimal value.

Chapter 7

Summary and Conclusions

This thesis explored two investment decisions in optical networks:

- the option to build new infrastructure,
- the option to increase capacity on an existing link.

Within this framework, we also considered two pertinent issues in the information communications technology (ICT) industry: decreasing investment costs and demand saturation.

In Chapter 3, we considered two fixed cost models for building new infrastructure. In the first model, a geometric Brownian motion (GBM) was used to model the value process. In the second model, a multiplicative jump-diffusion process (JDP) was used to model the value process. These models were previously solved using a Partial Differential Equation (PDE) approach. We provided an alternative derivation using martingale techniques. The binomial model was used to determine whether perpetual models are a good approximation for typical expiry dates. In an example from the literature, we found that the perpetual model is not a good approximation when the expiry date is less than 60 years. Stopping times were used to measure investment times. We found exact formulae for stopping time distributions in the GBM models and used simulation techniques to verify these results. In the JDP model, there is no exact formula for the stopping time distributions and so these must be estimated using simulation techniques. This makes the JDP models more difficult to study.

In Chapter 4, we extended the new infrastructure models to support decreasing investment costs. For large decay parameters we found that the GBM and JDP models were close and that the optimal trigger could be estimated using a linear approximation under these conditions. We also observed that the perpetual model was a good approximation for significantly lower expiry dates (e.g. 5 years instead of 60 years). Next, we developed an error model for the decay parameter and investigated various error scenarios. The model was found to be robust, since relatively large errors did not reduce the investment value by more than 5%. We also found that pessimistic predictions will never produce a negative payoff, but optimistic predictions may produce negative payoffs in some extreme cases. Finally, we developed an error model for traffic errors and presented a simple example. We found that there is no optimal trigger when the traffic parameter is large.

In Chapter 5, we presented a general strategy for deciding when to increase the link capacity and found an analytical solution for a GBM demand process. We showed that there is a unique optimal trigger that is greater than the initial link capacity, and found an explicit expression for the investment value. We also showed that the GBM model supported the special case of no prior investment.

In Chapter 6, we developed an increasing capacity model with demand saturation. A logistic process was used to model demand with saturation and Kummer's equation was used to find an analytical solution for the increasing capacity model. As in the GBM model, the optimal trigger is always greater than the initial link capacity, but we found that the optimal trigger may not exist when the investment cost is large. Since the investment value cannot be evaluated explicitly, we calculated upper bounds for the investment values or used simulation techniques to estimate the investment values. We found that the logistic model had lower investment values and later stopping times than the corresponding GBM model.

For each investment decision, we considered a GBM model first because it has exact formulae for many values. In the other models, simulation techniques were necessary to estimate these values. In each case, we observed that the GBM models were a good approximation for the more realistic JDP and logistic models under certain conditions (e.g. when the decay parameter or long-run equilibrium value were large). Future work may investigate this premise more systematically by developing error models for each case.

In Chapter 6, we mentioned that other models could have been used to model demand saturation (e.g. a logarithmic process) and our numerical results suggested that the logistic model had at most one optimal trigger. Future work may develop alternative demand saturation models to see whether they offer any further insights and determine whether this property is true in general.

There are several possible extensions to this thesis. It would be useful to fit real-world data to the models and calculate the real-world stopping time distributions, although real-world data is often difficult to obtain because of proprietary issues. While this thesis has focused on investment decisions in optical networks, future work may apply these techniques in related areas. These could include wireless networks, access pricing issues and other industries with decreasing costs or demand saturation.

Appendix A

Mathematical Theory

This appendix provides some of the mathematical theory used in this thesis: convergence tests and limits, differential equations, numerical techniques and statistical concepts.

A.1 Convergence Tests and Theorems

The following theorems were used to show that a function or series converges [63, 62].

Theorem A.1 (Lebesgue Theorem of Dominated Convergence).
If $\{f_n\}$ is a sequence of measurable functions, with $f_n \to f$ pointwise almost everywhere as $n \to \infty$, and $|f_n| \leq g$, $\forall n$, where g is integrable. Then f is integrable, and

$$\int f d\mu \equiv \lim_{n\to\infty} \int f_n d\mu.$$

Theorem A.2 (The Sandwich Theorem).
Let f, g and h be functions defined on $I\backslash\{a\}$ and suppose that

$$f(x) \leq g(x) \leq h(x)$$

and

$$L = \lim_{x\to a} f(x) = \lim_{x\to a} h(x)$$

then

$$\lim_{x \to a} g(x) = L.$$

This theorem is also known as the squeeze theorem or the pinching theorem.

Theorem A.3 (The Ratio Test).
If u_n is the nth term in a series and we let

$$L = \lim_{n \to \infty} \left| \frac{u_{n+1}}{u_n} \right|$$

then

- *If $L < 1$, the series converges.*
- *If $L > 1$, the series diverges.*
- *If $L = 1$, the series may converge or diverge*

The test is also known as d'Alembert ratio test or the Cauchy ratio test.

A.2 Differential Equations

A Partial Differential Equation (PDE) approach was used to solve the increasing capacity models in Chapters 5 and 6. The Euler-Cauchy equation was used to find homogeneous solutions in Section 5.3. Kummer's Equation was used to find homogeneous solutions in Section 6.3. The method of variation of parameters was used to find a particular solution for the non-homogeneous Kummer Equation in Section 6.3.

Theorem A.4 (Euler-Cauchy Equation [62]).
The homogeneous differential equation

$$x^2 \frac{d^2y}{dx^2} + \alpha x \frac{dy}{dx} + \beta y = 0,$$

has solutions

$$y = \begin{cases} c_1|x|^{r_1} + c_2|x|^{r_2}, & (\alpha-1)^2 > 4\beta; \\ (c_1 + c_2 \log|x|)|x|^a, & (\alpha-1)^2 = 4\beta; \\ |x|^a[c_1 \cos(b\log|x|) + c_2 \sin(b \log|x|)], & (\alpha-1)^2 < 4\beta, \end{cases}$$

where

$$\begin{aligned} r_1 &= \frac{1}{2}\left[1-\alpha+\sqrt{(\alpha-1)^2-4\beta}\right], \\ r_2 &= \frac{1}{2}\left[1-\alpha-\sqrt{(\alpha-1)^2-4\beta}\right], \\ a &= \frac{1}{2}(1-\alpha), \\ b &= \frac{1}{2}\sqrt{4\beta-(\alpha-1)^2}. \end{aligned}$$

Theorem A.5 (Kummer's Equation [1]).

The homogeneous differential equation

$$zg''(z) + (b-z)g'(z) - ag(z) = 0,$$

has two independent solutions $M(a,b,z)$ *and* $U(a,b,z)$, *where*

$$\begin{aligned} M(a,b,z) &= \sum_{n=0}^{\infty} \frac{(a)_n z^n}{(b)_n n!}, \\ (a)_n &= \Pi_{i=0}^{n}(a+i), \\ U(a,b,z) &= \frac{\pi}{\sin \pi b}\left[\frac{M(a,b,z)}{\Gamma(1+a-b)\Gamma(b)} - z^{1-b}\frac{M(1+a-b,2-b,z)}{\Gamma(a)\Gamma(2-b)}\right]. \end{aligned}$$

Theorem A.6 (Method of variation of parameters [51]).

Let $u_1(x)$ *and* $u_2(x)$ *be two linearly independent solutions of the homogeneous equation*

$$a_0(x)y''(x) + a_1(x)y'(x) + a_2(x)y(x) = 0.$$

A particular solution $y_p(x)$ *of*

$$a_0(x)y''(x) + a_1(x)y'(x) + a_2(x)y(x) = f(x),$$

has the following form

$$y_p(x) = a(x)u_1(x) + b(x)u_1(x),$$

where

$$a'(x) = -\frac{u_2(x)}{W(x)}\frac{f(x)}{a_0(x)},$$
$$b'(x) = \frac{u_1(x)}{W(x)}\frac{f(x)}{a_0(x)},$$
$$W(x) = u_1(x)u_2'(x) - u_2(x)u_1'(x).$$

A.3 Numerical Techniques

Three numerical techniques were used in this thesis: the bisection method[16], Newton's method[16] and Simpson's rule [1]. The bisection method and Newton's method were used to:

- Calculate the positive root in the jump-diffusion models in Chapters 3 and 4.
- Find the optimal triggers in Chapters 5 and 6.

Simpson's rule was used to:

- Calculate the conditional expected stopping time $E[\tau^*|\tau^* \leq t]$ using (3.18).
- Calculate $C = E\left[\int_0^\infty \min(D(s), S)e^{-rs}ds\right]$ directly using Lemma 5.9.
- Estimate $C = E\left[\int_0^\infty \min(D(s), S)e^{-rs}ds\right]$ using simulation techniques.

Note that the stopping conditions (i.e. **While** $(n < N)$) in Algorithms A.7 and A.8 can be replaced by a stopping condition based on tolerance (e.g. **While** $(|f(x_n) - f(x_{n-1})| < \epsilon)$ or **While** $(|x_n - x_{n-1}| < \epsilon)$).

Algorithm A.7 (Bisection method). *Given a function $f(x)$ continuous on the interval $[a_0, b_0]$ and such that $f(a_0)f(b_0) < 0$.*

- *Set* $n := 1$.
- *While* $(n < N)$

– *Set* $m := \frac{a_{n-1}+b_{n-1}}{2}$

– *If* $f(a_{n-1})f(m) \leq 0$, *set* $a_n := a_{n-1}$, $b_n = m$.

– *Otherwise, set* $a_n := m$, $b_n := b_{n-1}$.

– $n := n + 1$.

Algorithm A.8 (Newton's method). *Given* $f(x)$ *continuously differentiable and a point* x_0

- *Set* $n := 1$.
- *While* $(n < N)$.
  – *Set* $x_n := x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$.
  – *Set* $n := n + 1$.

Theorem A.9 (Simpson's Rule).

$$\int_a^b f(x)dx = \frac{h}{3}\{f_0+4(f_1+f_3+\ldots+f_{2n-1})+2(f_2+f_4+\ldots+f_{2n-2})+f_{2n}\}-\frac{nh^5}{90}f^{(4)}(\xi),$$

where

$$\begin{aligned} h &= \frac{b-a}{2n}, \\ f_i &= f(ih). \end{aligned}$$

A.4 Statistical Concepts

The following statistical concepts were used in this thesis: the cumulative normal distribution, normal variates and confidence intervals. The cumulative normal distribution was used to calculate the Black-Scholes formula (1.3) and the C-value, $C = E\left[\int_0^\infty \min(D(s), S)e^{-rs}ds\right]$, for the GBM increasing capacity model (Lemma 5.9). The polynomial approximation in Theorem A.10 was used to estimate the cumulative normal distribution. Normal variates are used to simulate any stochastic process that involves a standard Brownian motion, and therefore apply to every stochastic process discussed in this thesis. The polar method (Algorithm A.11) was used to generate normal variates. Theorem A.12 was used to calculate confidence intervals for the simulation results in this thesis. The upper 0.975 critical value for the standard normal distribution, $z_{0.975} = 1.96$, is used for a 95% confidence interval,

$$\left(\overline{X}(n) - 1.96\sqrt{\frac{S^2(n)}{n}}, \overline{X}(n) + 1.96\sqrt{\frac{S^2(n)}{n}}\right).$$

By maintaining the following counters:

$$\begin{aligned} \sigma_1 &= \sum_{i=1}^{n} X_i, \\ \sigma_2 &= \sum_{i=1}^{n} X_i^2, \end{aligned}$$

we were able to calculate the sample mean and variance on the fly:

$$\begin{aligned} \overline{X}(n) &= \frac{\sigma_1}{n}, \\ S^2(n) &= \frac{n}{n-1}\left(\frac{\sigma_1}{n} - \left(\frac{\sigma_1}{n}\right)^2\right). \end{aligned}$$

Theorem A.10. *If $N(x)$ is the cumulative normal distribution,*

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{z^2}{2}\right) dz,$$

then the following polynomial approximation gives six-decimal place accuracy [1]

$$N(x) \approx 1 - N'(x)(b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5),$$

where

$$\begin{aligned} t &= \frac{1}{1+px}, \\ p &= .2316419, \\ b_1 &= .319381530, \\ b_2 &= -.356563782, \\ b_3 &= 1.781477937, \\ b_4 &= -1.821255978, \\ b_5 &= 1.330274429. \end{aligned}$$

Algorithm A.11 (Polar method [42, page 491]).

To generate two $N(0,1)$ random variates:

1. *Generate U_1 and U_2 as IID $U(0,1)$, let $V_i = 2U_i - 1$ for $i = 1, 2$, and let $W = V_1^2 + V_2^2$.*

2. *If $W > 1$, go back to step 1. Otherwise, let $Y = \sqrt{(-2\log(W))/W}$, $X_1 = V_1 Y$, and $X_2 = V_2 Y$. Then X_1 and X_2 are IID $N(0,1)$ random variates.*

Theorem A.12 (Confidence Intervals [42, Chapter 4]).

If $X_1, X_2, \ldots, X_n$ are independent identically distributed random variables with

$$E[X] = \mu,$$

and

$$Var[X] = \sigma^2,$$

then the sample mean

$$\overline{X}(n) = \frac{1}{n}\sum_{i=1}^{n} X_i,$$

is an unbiased estimator for μ*, the sample variance*

$$S^2(n) = \frac{1}{n-1}\sum_{i=1}^{n}\left[X_i - \overline{X}(n)\right]^2$$

is an unbiased estimator for σ^2 *and an appropriate 100*$(1-\alpha)$*% confidence interval for* μ*, is given by*

$$\left(\overline{X}(n) - z_{1-\alpha/2}\sqrt{\frac{S^2(n)}{n}}, \overline{X}(n) + z_{1-\alpha/2}\sqrt{\frac{S^2(n)}{n}}\right),$$

where $z_{1-\alpha/2}$ *is the upper* $1-\alpha/2$ *critical value for the standard normal distribution.*

Appendix B

Mathematical Results

This appendix contains derivations for mathematical results used in the thesis: the risk-neutral pricing formula, Dixit and Pindyck's work on the fixed-cost GBM model, Lassila's work on the fixed-cost JDP model, and some key results for Chapters 3 and 6.

B.1 The risk-neutral pricing formula

Consider a portfolio of Δ shares of stock and $\$B$ dollars in riskless bonds and let $R = \exp(rt)$ denote one plus the risk-free rate. To avoid arbitrage between the stock and the riskless asset, we require $d < R < u$. The values of the portfolio at the two times and in the upstate and downstate are shown below.

$\Delta uS + RB$ with probability q,

$\Delta S + B$

$\Delta dS + RB$ with probability $1 - q$.

We may choose Δ and B in such a way that our portfolio replicates the value of the option in each state:

$$\begin{aligned} \Delta uS + RB &= V^+, \\ \Delta dS + RB &= V^-. \end{aligned}$$

Solving for Δ and B gives

$$\begin{aligned} \Delta &= \frac{V^+ - V^-}{(u-d)S}, \\ B &= \frac{uV^- - dV^+}{(u-d)R}. \end{aligned}$$

Provided there are no arbitrage opportunities, the value of the option should be equal to the value of its replicating portfolio:

$$\begin{aligned} V &= \Delta S + B \\ &= \frac{V^+ - V^-}{(u-d)S}S + \frac{uV^- - dV^+}{(u-d)R} \\ &= \frac{(\frac{R-d}{u-d})V^+ + (\frac{u-R}{u-d})V^-}{R}. \end{aligned} \tag{B.1}$$

Defining

$$p \equiv \frac{R-d}{u-d}, \tag{B.2}$$

we can rewrite Equation (B.1) as

$$V = \frac{pV^+ + (1-p)V^-}{R}. \tag{B.3}$$

Since p and $1-p$ are between zero and one, they can be regarded as probabilities. They are called the risk-neutral probabilities and (B.3) is called the risk-neutral pricing formula. Notice that the risk-neutral pricing formula is independent of the real world probabilities q and $(1-q)$.

A related formula applies when there are continuous dividends [33]. For continuous dividends δ, we let $A = \exp((r-\delta)t)$ and define

$$p \equiv \frac{A-d}{u-d}. \tag{B.4}$$

The risk-neutral value V is still given by (B.3).

B.2 Dixit and Pindyck's GBM Model

Dixit and Pindyck [22] considered a deterministic case of the investment model and then used dynamic programming and contingent claim analysis to solve the GBM model. This section provides an overview of their work.

The Investment Problem

McDonald and Siegel (1986) considered the following problem: At what point is it optimal to pay a sunk cost I in return for a project whose value is V, given that V evolves according to the following geometric Brownian motion:

$$dV = (\rho - \delta)V dt + \sigma V dB(t), \tag{B.5}$$

where ρ is the discount rate, δ is the dividend rate, σ is the volatility, and $B(t)$ is a standard Brownian motion. Dixit and Pindyck (1994) analysed the value of the investment opportunity,

$$F(V) = \max E[(V_T - I)e^{-\rho T}], \tag{B.6}$$

where E denotes the expectation, T is the (unknown) future time that the investment is made, ρ is a discount rate, and the maximization is subject to Equation (B.5) for V. We assume that $\delta \geq 0$, otherwise the intergral in Equation (B.5) could be made infinitely larger by choosing a larger T. Dixit and Pindyck first considered the deterministic case (i.e. $\sigma = 0$) and then used dynamic programming and contingent claims analysis to analyse the stochastic case. Their results are given in the following sections.

The Deterministic Case

If $\sigma = 0$, then $V(t) = Ve^{(\rho-\delta)t}$, where $V(0) = V$. Thus the value of the investment opportunity assuming we invest at some arbitrary future time T is

$$F(V) = (Ve^{(\rho-\delta)T} - I)e^{-\rho T}. \tag{B.7}$$

Suppose $\rho \leq \delta$. Then $V(t)$ will remain constant or fall over time, so the optimal strategy is to invest immediately if $V > I$, and never invest otherwise. Hence $F(V) = (V - I)^+$. Now consider $0 < \delta < \rho$. Then $F(V) > 0$ even if currently $V < I$, because eventually V will exceed I. Also even if V now exceeds I, it may be better to wait rather than invest now. Maximizing F(V) with respect to T, we get

$$\frac{dF(V)}{dT} = -(\delta)Ve^{-(\delta)T} + \rho I e^{-\rho T} = 0,$$

which implies

$$T^* = \max\left\{\frac{1}{\rho - \delta}\log\left[\frac{\rho I}{(\delta V}\right], 0\right\} \tag{B.8}$$

By setting $T^* = 0$, we see that one should invest immediately if $V \geq V^*$, where

$$V^* = \frac{\rho}{\delta} I > I. \tag{B.9}$$

Finally, by substituting Expression (B.8) into Equation (B.7), we obtain the following solution for F(V):

$$F(V) = \begin{cases} [(\rho - \delta)I/(\delta)][(\delta)V/\rho I]^{\rho/(\rho-\delta)}, & V \leq V^*; \\ V - I, & V > V^*. \end{cases} \tag{B.10}$$

The General Case by Dynamic Programming

We now consider the general case where $\sigma > 0$. The problem is to determine the point at which it is optimal to invest I in return for an asset worth V. The investment rule will take the form of a critical value V^* such that it is optimal to invest once $V \geq V^*$.

Because the investment opportunity, $F(V)$, yields no cash flows up to the time T that the investment is undertaken, the only return from holding it is its capital appreciation. Hence, in the continuation region (values of V for which it is not optimal to invest) the Bellman equation is

$$\rho F dt = E[dF]. \tag{B.11}$$

Equation (B.11) demonstrates that over a time interval dt, the total expected return on the investment opportunity, $\rho F dt$, is equal to its expected rate of capital appreciation.

We expand dF using Itô's lemma,

$$dF = F'(V)dV + \frac{1}{2}F''(V)(dV)^2.$$

Substituting Equation (B.5) for dV into this expression and noting that $E[dB(t)] = 0$ gives

$$E[dF] = (\rho - \delta)VF'(V)dt + \frac{1}{2}\sigma^2 V^2 F''(V)dt.$$

Hence the Bellman equation becomes the following differential equation (after dividing through by dt), this must be satisfied by $F(V)$:

$$\frac{1}{2}\sigma^2 V^2 F''(V) + (\rho - \delta)VF'(V) - \rho F = 0 \qquad \text{(B.12)}$$

In addition, $F(V)$ must satisfy the following boundary conditions:

$$F(0) = 0, \qquad \text{(B.13)}$$

$$F(V^*) = V^* - I, \qquad \text{(B.14)}$$

$$F'(V^*) = 1. \qquad \text{(B.15)}$$

To find $F(V)$, we must solve Equation (B.12) subject to the boundary conditions (B.13)-(B.15). To satisfy the boundary condition (B.13), the solution must take the form

$$F(V) = AV^k. \qquad \text{(B.16)}$$

Substituting this expression into Equation (B.12) and dividing through by AV^λ, we obtain the quadratic equation:

$$\frac{1}{2}\sigma^2 k(k-1) + (\rho - \delta)k - \rho = 0. \qquad \text{(B.17)}$$

This quadratic has two roots

$$k = \frac{1}{2} - \frac{\rho - \delta}{\sigma^2} + \sqrt{\left[\frac{\rho - \delta}{\sigma^2} - \frac{1}{2}\right]^2 + \frac{2\rho}{\sigma^2}} > 1,$$

$$k_2 = \frac{1}{2} - \frac{\rho - \delta}{\sigma^2} - \sqrt{\left[\frac{\rho - \delta}{\sigma^2} - \frac{1}{2}\right]^2 + \frac{2\rho}{\sigma^2}} < 0,$$

so the general solution to Equation (B.12) can be written as

$$F(V) = AV^{k} + BV^{k_2},$$

where A and B are constants to be determined. In our problem, the boundary condition (B.13) implies that $B = 0$, leaving the solution (B.16).

The remaining boundary conditions, (B.14) and (B.15), can be used to solve for the two remaining unknowns - the constant A, and the critical value V^* at which it is optimal to invest. By substituting (B.16) into (B.14) and (B.15) and rearranging, we find that

$$V^* = \frac{k}{k-1} I, \qquad \text{(B.18)}$$

$$A = \frac{V^* - I}{(V^*)^k}. \qquad \text{(B.19)}$$

The General Case by Contingent Claim Analysis

Let us now determine $F(V)$ using Contingent Claim Analysis. The contigent claim analysis requires one important assumption: stochastic changes in V must be spanned by existing assets in the economy.

Consider the following portfolio: Hold the option to invest, which is worth $F(V)$, and go short $n = F'(V)$ units of the project (or equivalently, of the asset or portfolio x that is perfectly correlated with V). The value of this portfolio $\phi = F - F'(V)V$. Note that this portfolio is dynamic; as V changes, $F'(V)$ may change from one short interval of time to the next, so that the composition of the portfolio will be changed. However, over each short interval of length dt, we hold n fixed.

The short position in this portfolio will require a payment of $\delta V F'(V)$ dollars per unit time period; otherwise no rational investor will enter into the long side of the transaction. An investor holding the long position in the project will demand the risk-adjusted return μV, which equals the capital gain $(\rho - \delta)V$ plus the dividend stream δV. Since the short position includes $F'(V)$ units of the project, it will require paying out $\delta V F'(V)$. Taking this payment into account, the total return from holding the portfolio over the short time interval dt is

$$dF - F'(V)dV - \delta V F'(V)dt.$$

To obtain an expression for dF, use Itô's lemma:

$$dF = F'(V)dV + \frac{1}{2}F''(V)(dV)^2.$$

Hence the total return on the portfolio is

$$\frac{1}{2}F''(V)(dV)^2 - \delta V F'(V)dt.$$

From Equation (B.5) for dV, we know that $(dV)^2 = \sigma^2 V^2 dt$ so the return on the portfolio becomes

$$\frac{1}{2}\sigma^2 V^2 F''(V)dt - \delta V F'(V)dt.$$

Note that this return is risk-free. Hence to avoid arbitrage possibilities, it must equal $r\phi dt = r[F - F'(V)V]dt$:

$$\frac{1}{2}\sigma^2 V^2 F''(V)dt - \delta V F'(V)dt = r[F - F'(V)V]dt.$$

Dividing through by dt and rearranging gives the following differential equation that $F(V)$ must satisfy:

$$\frac{1}{2}\sigma^2 V^2 F''(V) - (r - \delta)V F'(V) - rF = 0. \tag{B.20}$$

Observe that this equation is almost identical to Equation (B.12) obtained using dynamic programming. The only difference is that the risk-free interest rate r replaces the discount rate ρ. The same boundary conditions (B.13)-(B.15) will also apply here. Thus the solution for $F(V)$ again has the form

$$F(V) = AV^k,$$

except that now r replaces ρ in the quadratic equation for the exponent k. Therefore

$$k = \frac{1}{2} - \frac{r-\delta}{\sigma^2} + \sqrt{\left[\frac{r-\delta}{\sigma^2} - \frac{1}{2}\right]^2 + \frac{2r}{\sigma^2}}. \tag{B.21}$$

The critical value V^* and the constant A are again given by equations (B.18) and (B.19).

B.3 Lassila's JDP Model

Lassila [41] used Bellman equations to derive the JDP Model. This section provides an overview of his work.

The price process

The price process $P(t)$ is determined by assuming that

1. The price process P(t) is given by

$$P(t) = \frac{kD(t)}{S(t)} = f(D, S),$$

where $D(t)$ is the connection's demand, $S(t)$ is the connection's supply, and k is a positive scaling constant.

2. The demand process $D(t)$ follows a geometric Brownian motion with

$$dD(t) = \nu_D D(t)dt + \sigma_D D(t)dB_D(t).$$

3. The supply process $S(t)$ follows a combined process of a geometric Brownian motion and a discrete Poisson process with

$$dS(t) = \nu_S S(t)dt + \sigma_S S(t)dB_S(t) + \phi S(t)dN(t), \quad \text{(B.22)}$$

$$dB_S(t)dB_D(t) = \rho dt. \quad \text{(B.23)}$$

Applying Itô's lemma,

$$\begin{aligned}
dP &= \frac{\partial f}{\partial D} + \frac{\partial f}{\partial S} + \frac{1}{2}\left[\frac{\partial^2 f}{\partial D^2}(dD)^2 + 2\frac{\partial^2 f}{\partial D \partial S} dD dS + \frac{\partial^2 f}{\partial S^2}(dS)^2\right] \\
&= \frac{k}{S} dD + \frac{-kD}{S^2} dS + \frac{1}{2}\left[2\frac{-k}{S^2} dD dS + \frac{2kD}{s^3}(dS)^2\right] \\
&= \frac{k}{S} dD - \frac{kD}{S^2} dS - \frac{k}{S^2} dD dS + \frac{kD}{S^3}(dS)^2 \\
&= \frac{k}{S}\left(\alpha_D D(t)dt + \sigma_D D(t) dB_D(t)\right) - \frac{kD}{S^2}\left(\alpha_S S(t)dt + \sigma_S S(t) dB_S(t) + \phi S(t) dN(t)\right) \\
&\quad -\frac{k}{S^2}\left(\alpha_D D(t)dt + \sigma_D D(t) dB_D(t)\right) \times \left(\alpha_S S(t)dt + \sigma_S S(t) dB_S(t) + \phi S(t) dN(t)\right) \\
&\quad +\frac{kD}{S^3}\left(\alpha_S S(t)dt + \sigma_S S(t) dB_S(t) + \phi S(t) dN(t)\right)^2 \\
&= \frac{kD}{S}\left[(\alpha_D - \alpha_S + \sigma_S^2 - \rho\sigma_D\sigma_S)dt + \sigma_D dB_D(t) - \sigma_S dB_S(t) - \phi dN(t)\right] \\
&= (\alpha_D - \alpha_S + \sigma_S^2 - \rho\sigma_D\sigma_S)P(t)dt + \sigma_D P(t) dB_D(t) - \sigma_S P(t) dB_S(t) - \phi P(t) dN(t).
\end{aligned}$$

The expression for dP can be re-written as

$$dP(t) = \nu_P P(t)dt + \sigma_P P(t) dB_P(t) - \phi P(t) dN(t), \tag{B.24}$$

where

$$\begin{aligned}
\nu_P &= \alpha_D - \alpha_S + \sigma_S^2 - \rho\sigma_D\sigma_S, \\
\sigma^2 &= \sigma_D^2 - 2\rho\sigma_D\sigma_S + \sigma_D^2, \\
B_P(t) &\equiv \frac{\sigma_D B_D(t) - \sigma_S B_S(t)}{\sqrt{\sigma_D^2 - 2\rho\sigma_D\sigma_S + \sigma_S^2}}.
\end{aligned}$$

By the Lévy theorem, B_P is also a Brownian motion.

The present value of income

The present value of income V_T is the value of an investment at time T, and is given by

$$V_T = E\left[\int_{T+\tau}^{\infty} P(t) e^{-r(t-T)} dt\right]. \tag{B.25}$$

This model assumes that there is a delay, τ, between construction and operation and that the carrier is risk-neutral. The expected instantaneous growth rate of Equation

(B.24) is

$$E[dP(t)] = (\nu - \phi\lambda)P(t)dt,$$

where λ is the Poisson arrival rate. This equation has a solution

$$E[P(t)] = P(0)e^{\nu \, t},$$

where $\nu' = \nu - \phi\lambda$. Substituting this solution into the expression for the present value of income we get

$$\begin{aligned} V_T &= E\left[\int_{T+\tau}^{\infty} P(t)e^{-r(t-T)}dt\right] \\ &= E\left[\int_{0}^{\infty} P(t+T+\tau)e^{-r(t+T+\tau-T)}dt\right] \\ &= \int_{0}^{\infty} E[P(t+T+\tau)]e^{-r(t+\tau)}dt \\ &= \int_{0}^{\infty} P(T)e^{\nu \, (t+\tau)}e^{-r(t+\tau)}dt \\ &= P(T)e^{-(r-\nu \,)\tau}\int_{0}^{\infty} e^{(\nu \, -r)t}dt \\ &= P(T)e^{-(r-\nu \,)\tau}\frac{-1}{\nu' - r} \\ &= e^{-(r-\nu \,)\tau}\frac{P(T)}{r - \nu'}. \end{aligned} \tag{B.26}$$

Remarks: The result requires that $\nu' < r$. Otherwise the integral is unbounded. Observe that V_T follows the same process as the spot price P(t) and so

$$dV(t) = \nu V(t)dt + \sigma V(t)dB(t) - \phi V(t)dN(t).$$

Real Option Analysis

The value of the investment opportunity $F(V)$ is

$$F(V) = \max E[(V(T) - X)e^{-rT}].$$

The Bellman Equation is

$$rFdt = E[dF(V)].$$

The $dF(V)$ term can be expanded using Itô's lemma

$$dF(V) = \frac{\partial F(V)}{\partial V} dV + \frac{1}{2}\frac{\partial^2 F(V)}{\partial V^2}(dV)^2. \tag{B.27}$$

Expanding the $(dV)^2$ term, we get

$$\begin{aligned}(dV)^2 &= \mu^2 V^2 (dt)^2 + 2\mu\sigma V^2 dt dB(t) - 2\phi\mu V^2 dt dN(t) + \sigma V^2 (dB)^2 \\ &\quad -2\phi\sigma V^2 dB(t) dN(t) + \phi^2 V^2 (dN(t))^2. \\ &= \sigma^2 V^2 dt.\end{aligned}$$

Using primes to denote derivatives, the $dF(V)$ term becomes

$$dF(V) = F'(V)[\nu V dt + \sigma V dB(t) - \phi V dN(t)] + \frac{1}{2} F''(V)\sigma^2 V^2 (dB)^2. \tag{B.28}$$

Consider the expected value of the jump term,

$$\begin{aligned}E\left[-\frac{\partial F(V)}{\partial V}\phi V dN(t)\right] &= -\frac{\partial F(V)}{\partial V}\phi V p(\text{jump}) - \frac{\partial F(V)}{\partial V}\phi V p(\text{no jump}) \\ &= -\frac{F(V) - F((1-\phi))}{V - (1-\phi)V}\phi V \beta dt \\ &= \frac{F((1-\phi)V) - F(V)}{\phi V}\phi V \beta dt \\ &= \lambda[F((1-\phi)V) - F(V)]dt. \end{aligned} \tag{B.29}$$

Using Equations (B.28) and (B.29), and observing that $E[dW] = 0$, the Bellman equation becomes

$$\frac{1}{2}\sigma^2 V^2 F''(V) + \nu V F'(V) - (\lambda + r)F(V) + \lambda F((1-\phi)V) = 0. \tag{B.30}$$

In addition, $F(V)$ must satisfy the following boundary conditions

$$F(0) = 0, \tag{B.31}$$

$$F(V^*) = V^* - I, \tag{B.32}$$

$$F'(V^*) = 1. \tag{B.33}$$

To satisfy the boundary condition (B.31), the solution must take the form

$$F(V) = AV^k.$$

Substituting this expression into the Bellman equation and dividing through by AV^{λ} yields

$$\frac{1}{2}\sigma^2 k(k-1) + \nu k - (\lambda + r) + \lambda(1-\phi)^k = 0. \tag{B.34}$$

Boundary condition (B.31) implies that k has to be positive. The solution to Equation (B.34) must be obtained numerically. If there are no jumps we set $\lambda := 0$ and

$$k = \frac{1}{2} - \frac{\nu}{\sigma^2} + \sqrt{\left[\frac{\nu}{\sigma^2} - \frac{1}{2}\right]^2 + \frac{2r}{\sigma^2}}.$$

The remaining boundary conditions (B.32) and (B.33) can be used to solve the remaing unknowns - the constant A, and the critical value V^* at which it is optimal to invest.

$$\begin{aligned} V^* &= \frac{k}{k-1} I, \\ A &= \frac{V^* - I}{V^{*k}}. \end{aligned}$$

The option should be exercised when $V > V^*$.

B.4 Results for Chapter 3

A martingale approach was used to derive the fixed-cost models in Chapter 3. Theorems B.1 and B.2 were used to derive the fixed-cost GBM model. Theorems B.3 and B.4 were used to derive the fixed-cost JDP model.

Theorem B.1. *The geometric Brownian motion defined by*

$$dY = \nu Y dt + \sigma Y dB(t),$$

has an exact solution

$$Y(t) = Y_0 \exp\left\{\left(\nu - \frac{\sigma^2}{2}\right) t + \sigma B(t)\right\}.$$

Proof. Define $X(Y,t) = \log Y(t)$, then by Itô's lemma,

$$\begin{aligned} dX &= \frac{\partial X}{\partial t}dt + \frac{\partial X}{\partial Y}dY + \frac{\partial^2 X}{\partial Y^2}(dY)^2 \\ &= \frac{1}{Y}dY + \frac{1}{2}(-\frac{1}{Y^2})\sigma^2 Y^2 dt \\ &= \frac{1}{Y}(\nu Y dt + \sigma Y dB(t)) - \frac{\sigma^2}{2}dt \\ &= \left(\nu - \frac{\sigma^2}{2}\right) dt + \sigma dB(t). \end{aligned}$$

Thus

$$d(\log Y) = \left(\nu - \frac{\sigma^2}{2}\right) dt + \sigma dB(t).$$

Integrating leads to

$$\begin{aligned} \log Y]_0^t &= \left(\nu - \frac{\sigma^2}{2}\right)\tau\Big]_0^t + \sigma B(\tau)]_0^t \\ \log Y(t) - \log Y_0 &= \left(\nu - \frac{\sigma^2}{2}\right) t + \sigma B(t) \\ \log(Y(t)/Y_0) &= \left(\nu - \frac{\sigma^2}{2}\right) t + \sigma B(t), \end{aligned}$$

rearranging we get

$$Y(t) = Y(0)\exp\left[\left(\nu - \frac{\sigma^2}{2}\right) t + \sigma B(t)\right].$$

□

Theorem B.2. *The process* $\{\exp(k\sigma B(t) - \frac{1}{2}k^2\sigma^2 t) : t \leq 0\}$ *is a martingale.*

Proof. Let $X(t) = \exp(\sigma B(t) - \frac{1}{2}\sigma^2 t)$.

$$\begin{aligned} Y(t) - Y(0) &= \int_0^t X(s)\left[\sigma dB(s) - \frac{1}{2}\sigma^2 ds\right] + \frac{1}{2}\int_0^t X(s)\sigma^2 ds \\ &= \int_0^t X(s)\sigma dB(s). \end{aligned}$$

This expression is a martingale because $E[\int_0^t X(s)\sigma dB(s)] = 0$. □

Theorem B.3. *The jump-diffusion process defined by*

$$dY(t) = \nu Y(t)dt + \sigma Y(t)dB(t) - \phi Y(t)dN(t)$$

has an exact solution $Y(t) = e^{X_t}$ *where*

$$X_t = \left(\nu - \frac{\sigma^2}{2}\right)t + \sigma B(t) + \ln(1-\phi)N(t).$$

Proof. By Itô's lemma,

$$\begin{aligned} Y(t) - Y(0) &= \int_{0+}^{t} e^{X_{s-}}dX_s + \frac{1}{2}\int_{0+}^{t} e^{X_{s-}}\sigma^2 ds + \sum_{0<s\leq t}\{e^{X_s} - e^{X_{s-}} - e^{X_{s-}}\Delta X_s\} \\ &= \int_{0+}^{t} e^{X_{s-}}dX_s + \frac{1}{2}\int_{0+}^{t} e^{X_{s-}}\sigma^2 ds \\ &+ \sum_{0<s\leq t}\{e^{X_{s-}+\mu} - e^{X_{s-}} - e^{X_{s-}}\mu\}I(\Delta X_s \neq 0) \\ &= \int_{0+}^{t} e^{X_{s-}}dX_s + \frac{1}{2}\int_{0+}^{t} e^{X_{s-}}\sigma^2 dt \\ &\quad + \sum_{0<s\leq t} e^{X_{s-}}[e^{\mu} - 1 - \mu]I(\Delta X_s \neq 0) \end{aligned}$$

Taking derivatives

$$dY(t) = e^{X_{t-}}dX_t + \frac{1}{2}e^{X_{t-}}\sigma^2 dt + e^{X_t}[e^{\mu} - 1 - \mu]I(\Delta X_t \neq 0),$$

where $\Delta X_t = \mu dN(t) = \ln(1-\phi)\Delta N(t)$. Observing that $dN(t) = I(\Delta X_t \neq 0)$,

$$\begin{aligned} dY(t) &= Y(t-)\left[dX_t + \frac{\sigma^2}{2}dt + (e^{\ln(1-\phi)} - 1 - \ln(1-\phi))dN(t)\right] \\ &= Y(t-)\left[dX_t + \frac{\sigma^2}{2}dt + ((1-\phi) - 1 - \ln(1-\phi))dN(t)\right] \\ &= Y(t-)\left[dX_t + \frac{\sigma^2}{2}dt + (-\phi - \ln(1-\phi))dN(t)\right]. \end{aligned} \tag{B.35}$$

To complete the proof, we let

$$X_t = \left(\nu - \frac{\sigma^2}{2}\right)t + \sigma B(t) + \ln(1-\phi)N(t), \tag{B.36}$$

and show that

$$dY(t) = \nu Y(t)dt + \sigma Y(t)dB(t) - \phi Y(t)dN(t)$$

is satisfied. Substituting (B.36) into (B.35),

$$\begin{aligned} dY(t) &= Y(t-)\left[\left(\nu - \frac{\sigma^2}{2}\right) dt + \sigma dB(t) + \ln(1-\phi)dN(t) \right. \\ &\quad \left. + \frac{\sigma^2}{2}dt + (-\phi - \ln(1-\phi))dN(t)\right] \\ &= Y(t-)\left[\nu dt + \sigma dB(t) - \phi dN(t)\right]. \end{aligned}$$

□

Theorem B.4. *If* $g(k) = \frac{\sigma^2}{2}k(k-1) + \nu k + \lambda((1-\phi)^k - 1)$, *then the process* $\exp\{kX_t - g(k)t\}$ *is a martingale.*

Proof. Let $Z(t) = \exp[kX_t - g(k)t]$.

$$\begin{aligned} Z(t) - Z(0) &= \int_{0+}^t Z(s-)[kdX_s - g(k)ds] + \frac{1}{2}\int_{0+}^t Z(s-)k^2\sigma^2 ds \\ &\quad + \int_{0+}^t Z(s-)[e^{k\ln(1-\phi)} - 1 - k\ln(1-\phi)]dN(s) \\ &= \int_{0+}^t Z(s-)\left[k\left(\nu - \frac{\sigma^2}{2}\right)ds + k\sigma dB(s) + k\ln(1+\phi)dN(s) - g(k)ds\right] \\ &\quad + \frac{1}{2}\int_{0+}^t Z(s-)k^2\sigma^2 ds + \int_{0+}^t Z(s-)[(1-\phi)^k - 1 - k\ln(1-\phi)]dN(s) \\ &= \int_{0+}^t Z(s-)\left(k\nu + \frac{\sigma^2}{2}k(k-1) - g(k)\right)ds \\ &\quad + \int_{0+}^t Z(s-)k\sigma dB(s) + + \int_{0+}^t Z(s-)[(1-\phi)^k - 1]dN(s) \\ &= \int_{0+}^t Z(s-)\left(k\nu + \frac{\sigma^2}{2}k(k-1) + \lambda[(1-\phi)^k - 1] - g(k)\right)ds \\ &\quad + \int_{0+}^t Z(s-)k\sigma dB(s) + \int_{0+}^t Z(s-)[(1-\phi)^k - 1][dN(s) - \lambda ds]. \end{aligned}$$

This expression is a martingale when $g(k) = k\nu + \frac{\sigma^2}{2}k(k-1) + \lambda[(1-\phi)^k - 1]$ because $E[\int_{0+}^t Z(s-)k\sigma dB(s)] = 0$ and $E[\int_{0+}^t Z(s-)[(1-\phi)^k - 1][dN(s) - \lambda ds]] = 0$. □

B.5 Results for Chapter 6

A Partial Differential Equation (PDE) approach was used to derive the logistic model in Chapter 6. The following results were used in this derivation.

Derivatives

The following lemmas provide derivatives for functions involving the Kummer function. Applying Lemmas B.5-B.7, we note that the ψ functions defined in Section 6.3 have the following derivatives:

$$\begin{aligned}
\frac{\partial\psi_1(x)}{\partial x} &= ax^{a-1}M\left(a+1,b,\frac{2\eta x}{\sigma^2}\right),\\
\frac{\partial\psi_2(x)}{\partial x} &= (a+1-b)x^{a-b}M\left(a+2-b,2-b,\frac{2\eta x}{\sigma^2}\right),\\
\frac{\partial\psi_3(x)}{\partial x} &= a_1x^{a_1-1}M\left(a_1+1,b_1,\frac{2\eta x}{\sigma^2}\right),\\
\frac{\partial\psi_4(x)}{\partial x} &= a(1+a-b)x^{a-1}U\left(a+1,b,\frac{2\eta x}{\sigma^2}\right),\\
\frac{\partial\psi_5(x)}{\partial x} &= \frac{2\eta}{\sigma^2}\frac{a}{b}M\left(a+1,b,\frac{2\eta x}{\sigma^2}\right),\\
\frac{\partial\psi_6(x)}{\partial x} &= \frac{2\eta}{\sigma^2}(1-b)\left(\frac{2\eta x}{\sigma^2}\right)^{-b}M\left(a+1-b,1-b,\frac{2\eta x}{\sigma^2}\right),\\
\frac{\partial\psi_7(x)}{\partial x} &= \frac{2\eta}{\sigma^2}\left(\frac{2\eta x}{\sigma^2}\right)^{-a}M\left(1-a,2-b,-\frac{2\eta x}{\sigma^2}\right),\\
\frac{\partial\psi_8(x)}{\partial x} &= \frac{2\eta}{\sigma^2}\left(\frac{2\eta x}{\sigma^2}\right)^{b-a-1}M\left(b-\theta,b,-\frac{2\eta x}{\sigma^2}\right),\\
\frac{\partial\psi_9(x)}{\partial x} &= \frac{2\eta}{\sigma^2}\left(\frac{2\eta x}{\sigma^2}\right)^{-a-1}M\left(1-\theta,2-b,\frac{-2\eta x}{\sigma^2}\right),\\
\frac{\partial\psi_{10}(x)}{\partial x} &= \frac{2\eta}{\sigma^2}\left(\frac{2\eta x}{\sigma^2}\right)^{b-a-2}M\left(b-a,b,\frac{-2\eta x}{\sigma^2}\right).
\end{aligned}$$

Lemma B.5.

$$\frac{\partial x^a M(a,b,x)}{\partial x} = ax^{a-1}M(a+1,b,x).$$

Proof (1). Using Equation 13.4.10 from Abramowitz and Stegan [1],

$$\begin{aligned}\frac{\partial x^a M(a,b,x)}{\partial x} &= a x^{a-1} M(a,b,x) + x^a M'(a,b,x) \\ &= x^{a-1} a M(a+1,b,x).\end{aligned}$$

□

Proof (2). From first principles,

$$\begin{aligned}\frac{\partial x^a M(a,b,x)}{\partial x} &= \frac{\partial \sum_{n=0}^{\infty} \frac{(a)_n}{(b)_n} \frac{x^{a+n}}{n!}}{\partial x} \\ &= \sum_{n=0}^{\infty} \frac{(a)_n}{(b)_n} (a+n) \frac{x^{a+n-1}}{n!} \\ &= a x^{a-1} \sum_{n=0}^{\infty} \frac{(a+1)_n}{(b)_n} \frac{x^n}{n!} \\ &= a x^{a-1} M(a+1,b,x).\end{aligned}$$

□

Lemma B.6.

$$\frac{\partial x^{-a} M(-a,b,-x)}{\partial x} = -a x^{-a-1} M(-a+1,b,-x).$$

Proof. Using Equation 13.4.10 from [1],

$$\begin{aligned}\frac{\partial x^{-a} M(-a,b,-x)}{\partial x} &= -a x^{-a-1} M(-a,b,-x) + x^a M'(-a,b,-x)(-1) \\ &= -a x^{-a-1} M(-a+1,b,-x).\end{aligned}$$

□

Lemma B.7.

$$\frac{\partial x^a M(a,b,kx)}{\partial x} = a x^{a-1} M\left(a+1,b,kx\right) \;\; \forall k > 0.$$

Proof. Using Lemma B.5,

$$\begin{aligned}\frac{\partial x^a M(a,b,kx)}{\partial x} &= \frac{k^{-a}(kx)^a M(a,b,kx)}{\partial kx}\frac{\partial kx}{\partial x} \\ &= k^{-a}a(kx)^{a-1}M(a+1,b,kx)\,k \\ &= k^{-a+1}a(kx)^{a-1}M(a+1,b,kx) \\ &= ax^{a-1}M(a+1,b,kx)\,.\end{aligned}$$

□

Lemma B.8.

$$\frac{\partial x^{b-1}M(a,b,x)}{\partial x} = (b-1)x^{b-2}M(a,b-1,x)\,.$$

Proof (1). Using Equation 13.4.13 from [1],

$$\begin{aligned}\frac{\partial x^{b-1}M(a,b,x)}{\partial x} &= (b-1)x^{b-2}M(a,b,x) + x^a M'(a,b,x) \\ &= (b-1)x^{b-2}M(a,b-1,x).\end{aligned}$$

□

Proof (2). From first principles,

$$\begin{aligned}\frac{\partial x^{b-1}M(a,b,x)}{\partial x} &= \frac{\partial \sum_{n=0}^{\infty}\frac{(a)_n}{(b)_n}\frac{x^{b-1+n}}{n!}}{\partial x} \\ &= \sum_{n=0}^{\infty}\frac{(a)_n}{(b)_n}(b-1+n)\frac{x^{b+n-2}}{n!} \\ &= (b-1)x^{b-2}\sum_{n=0}^{\infty}\frac{(a)_n}{(b-1)_n}\frac{x^n}{n!} \\ &= (b-1)x^{b-2}M(a,b-1,x).\end{aligned}$$

□

Lemma B.9.

$$\frac{\partial x^a U(a,b,x)}{\partial x} = a(1+a-b)x^{a-1}U(a+1,b,x).$$

Proof. Using Equation 13.4.23 from [1],

$$\begin{aligned}\frac{\partial x^a U(a,b,x)}{\partial x} &= ax^{a-1}U(a,b,x) + x^a U'(a,b,x) \\ &= a(1+a-b)x^{a-1}U(a+1,b,x).\end{aligned}$$

□

Lemma B.10.

$$\frac{\partial x^a U(a,b,kx)}{\partial x} = a(1+a-b)x^{a-1}U(a+1,b,kx)\ \forall k > 0.$$

Proof. Using Lemma B.5,

$$\begin{aligned}\frac{\partial x^a U(a,b,kx)}{\partial x} &= \frac{k^{-a}(kx)^a U(a,b,kx)}{\partial kx}\frac{\partial kx}{\partial x} \\ &= k^{-a}a(1+a-b)(kx)^{a-1}U(a+1,b,kx)k \\ &= k^{-a+1}a(1+a-b)(kx)^{a-1}U(a+1,b,kx) \\ &= a(1+a-b)x^{a-1}U(a+1,b,kx).\end{aligned}$$

□

Lemma B.11. *If*

$$Y(a,b,x) = \sum_{n=0}^{\infty} \frac{(a)_n}{(b)_n}\frac{x^n}{(a+n)n!},$$

then

$$\frac{\partial x^a Y(a,b,x)}{\partial x} = x^{a-1}M(a,b,x).$$

Proof. From first principles,

$$\begin{aligned}\frac{\partial x^a Y(a,b,x)}{\partial x} &= \frac{\partial \sum_{n=0}^{\infty} \frac{(a)_n}{(b)_n}\frac{x^{a+n}}{(a+n)n!}}{\partial x} \\ &= \sum_{n=0}^{\infty} \frac{(a)_n}{(b)_n}(a+n)\frac{x^{a+n-1}}{(a+n)n!} \\ &= x^{a-1}\sum_{n=0}^{\infty} \frac{(a)_n}{(b)_n}\frac{x^n}{n!} \\ &= x^{a-1}M(a,b,x).\end{aligned}$$

□

Lemma B.12.
$$\frac{\partial x^{-a}Y(-a,b,kx)}{\partial x} = x^{-a-1}M(-a,b,kx).$$

Proof. From first principles,

$$\begin{aligned}
\frac{\partial x^{-a}Y(-a,b,kx)}{\partial x} &= \frac{\partial \sum_{n=0}^{\infty}(k)^n \frac{(-a)_n}{(b)_n}\frac{x^{n-a}}{(n-a)n!}}{\partial x} \\
&= \sum_{n=0}^{\infty}(k)^n \frac{(-a)_n}{(b)_n}(n-a)\frac{x^{n-a-1}}{(n-a)n!} \\
&= x^{-a-1}\sum_{n=0}^{\infty}\frac{(-a)_n}{(b)_n}\frac{(kx)^n}{n!} \\
&= x^{-a-1}M(-a,b,kx).
\end{aligned}$$

□

Lemma B.13. *If*

$$\Phi(x) := \psi_2(x)x^{b-a}Y\left(b-a,b,\frac{-2\eta x}{\sigma^2}\right) - \psi_1(x)x^{1-a}Y\left(1-a,2-b,\frac{-2\eta x}{\sigma^2}\right),$$

then

$$\Phi'(x) = \psi_2'(x)x^{b-a}Y\left(b-a,b,\frac{-2\eta x}{\sigma^2}\right) - \psi_1'(x)x^{1-a}Y\left(1-a,2-b,\frac{-2\eta x}{\sigma^2}\right).$$

Proof.

$$\begin{aligned}
\phi'(x) &= \frac{\partial}{\partial x}\left\{xM\left(1+a-b,2-b,\frac{2\eta x}{\sigma^2}\right)Y\left(b-a,b,\frac{-2\eta x}{\sigma^2}\right)\right\} \\
&\quad -\frac{\partial}{\partial x}\left\{xM\left(a,b,\frac{2\eta x}{\sigma^2}\right)Y\left(1-a,2-b,\frac{-2\eta x}{\sigma^2}\right)\right\} \\
&= \frac{\partial}{\partial x}\left\{x^{1+a-b}M\left(1+a-b,2-b,\frac{2\eta x}{\sigma^2}\right)x^{b-a}Y\left(b-a,b,\frac{-2\eta x}{\sigma^2}\right)\right\} \\
&\quad -\frac{\partial}{\partial x}\left\{x^{a}M\left(a,b,\frac{2\eta x}{\sigma^2}\right)x^{1-a}Y\left(1-a,2-b,\frac{-2\eta x}{\sigma^2}\right)\right\} \\
&= (1+a-b)x^{a-b}M\left(2+a-b,2-b,\frac{2\eta x}{\sigma^2}\right)x^{b-a}Y\left(b-a,b,\frac{-2\eta x}{\sigma^2}\right) \\
&\quad x^{1+a-b}M\left(1+a-b,2-b,\frac{2\eta x}{\sigma^2}\right)x^{b-a}M\left(b-a,b,\frac{-2\eta x}{\sigma^2}\right) \\
&\quad -ax^{a-1}M\left(a+1,b,\frac{2\eta x}{\sigma^2}\right)x^{1-a}Y\left(1-a,2-b,\frac{-2\eta x}{\sigma^2}\right) \\
&\quad -x^{a}M\left(a,b,\frac{2\eta x}{\sigma^2}\right)x^{1-a}M\left(1-a,2-b,\frac{-2\eta x}{\sigma^2}\right).
\end{aligned}$$

Applying the Kummer transformation (Equation 13.1.27 from [1]),

$$\begin{aligned}
&= (1+a-b)x^{a-b}M\left(2+a-b,2-b,\frac{2\eta x}{\sigma^2}\right)x^{b-a}Y\left(b-a,b,\frac{-2\eta x}{\sigma^2}\right) \\
&\quad x^{1+a-b}M\left(1+a-b,2-b,\frac{2\eta x}{\sigma^2}\right)x^{b-a}\exp\left(-\frac{2\eta x}{\sigma^2}\right)M\left(a,b,\frac{2\eta x}{\sigma^2}\right) \\
&\quad -ax^{a-1}M\left(a+1,b,\frac{2\eta x}{\sigma^2}\right)x^{1-a}Y\left(1-a,2-b,\frac{-2\eta x}{\sigma^2}\right) \\
&\quad -x^{a}M\left(a,b,\frac{2\eta x}{\sigma^2}\right)x^{1-a}\exp\left(-\frac{2\eta x}{\sigma^2}\right)M\left(1+a-b,2-b,\frac{2\eta x}{\sigma^2}\right) \\
&= (1+a-b)x^{a-b}M\left(2+a-b,2-b,\frac{2\eta x}{\sigma^2}\right)x^{b-a}Y\left(b-a,b,\frac{-2\eta x}{\sigma^2}\right) \\
&\quad -ax^{a-1}M\left(a+1,b,\frac{2\eta x}{\sigma^2}\right)x^{1-a}Y\left(1-a,2-b,\frac{-2\eta x}{\sigma^2}\right) \\
&= \psi_2'(x)x^{b-a}Y\left(b-a,b,\frac{-2\eta x}{\sigma^2}\right)-\psi_1(x)x^{1-a}Y\left(1-a,2-b,\frac{-2\eta x}{\sigma^2}\right).
\end{aligned}$$

□

Wronskians

Combining Lemma B.14 with (6.2), (6.3) and (6.4) enables us to find the Wronskians of key ψ functions.

Lemma B.14.

$$W(x^a u(x), x^a v(x)) = x^{2a} W(u(x), v(x)).$$

Proof.

$$\begin{aligned}
W(x^a u(x), x^a v(x)) &= x^a u(x)(ax^{a-1}v(x) + x^a v'(x)) - x^a v(x)(ax^{a-1}u(x) + x^a u'(x)) \\
&= x^{2a}(u(x)v'(x) - u'(x)v(x)) \\
&= x^{2a} W(u(x), v(x)).
\end{aligned}$$

□

Lemma B.15.

$$W(\psi_1, \psi_2) = (1-b)x^{2a-b} \exp\left(\frac{2\eta x}{\sigma^2}\right).$$

Proof. Using Lemma B.14,

$$\begin{aligned}
W(\psi_1, \psi_2) &= x^{2a} W\left(M\left(a, b, \frac{2\eta x}{\sigma^2}\right), x^{1-b} M\left(a+1-b, 2-b, \frac{2\eta x}{\sigma^2}\right)\right) \\
&= x^{2a} M\left(a, b, \frac{2\eta x}{\sigma^2}\right) \frac{\partial x^{1-b} M\left(a+1-b, 2-b, \frac{2\eta x}{\sigma^2}\right)}{\partial x} \\
&\quad - x^{2a} x^{1-b} M\left(a+1-b, 2-b, \frac{2\eta x}{\sigma^2}\right) \frac{\partial M\left(a, b, \frac{2\eta x}{\sigma^2}\right)}{\partial x} \\
&= x^{2a}\left(\frac{2\eta}{\sigma^2}\right)^{b-1} M\left(a, b, \frac{2\eta x}{\sigma^2}\right) \frac{\partial \left(\frac{2\eta x}{\sigma^2}\right)^{1-b} M\left(a+1-b, 2-b, \frac{2\eta x}{\sigma^2}\right)}{\partial \frac{2\eta x}{\sigma^2}} \frac{\partial \frac{2\eta x}{\sigma^2}}{\partial x} \\
&\quad - x^{2a}\left(\frac{2\eta}{\sigma^2}\right)^{b-1} x^{1-b} M\left(a+1-b, 2-b, \frac{2\eta x}{\sigma^2}\right) \frac{\partial M\left(a, b, \frac{2\eta x}{\sigma^2}\right)}{\partial \frac{2\eta x}{\sigma^2}} \frac{\partial \frac{2\eta x}{\sigma^2}}{\partial x} \\
&= x^{2a}\left(\frac{2\eta}{\sigma^2}\right)^{b} W\left(y_1\left(\frac{2\eta x}{\sigma^2}\right), y_2\left(\frac{2\eta x}{\sigma^2}\right)\right).
\end{aligned}$$

Using (6.2),

$$W(\psi_1, \psi_2) = (1-b)x^{2a-b}\exp\left(\frac{2\eta x}{\sigma^2}\right).$$

□

Lemma B.16.

$$W(\psi_1, \psi_4) = -\frac{\Gamma(b)}{\Gamma(a)}x^{2a-b}\exp\left(\frac{2\eta x}{\sigma^2}\right)\left(\frac{2\eta}{\sigma^2}\right)^{1-b}.$$

Proof. Using Lemma B.14,

$$\begin{aligned}
W(\psi_1, \psi_4) &= x^{2a}W\left(M\left(a, b, \frac{2\eta x}{\sigma^2}\right), U\left(a, b, \frac{2\eta x}{\sigma^2}\right)\right) \\
&= x^{2a}M\left(a, b, \frac{2\eta x}{\sigma^2}\right)\frac{\partial U\left(a, b, \frac{2\eta x}{\sigma^2}\right)}{\partial x} \\
&\quad -x^{2a}U\left(a, b, \frac{2\eta x}{\sigma^2}\right)\frac{\partial M\left(a, b, \frac{2\eta x}{\sigma^2}\right)}{\partial x} \\
&= x^{2a}M\left(a, b, \frac{2\eta x}{\sigma^2}\right)\frac{\partial U\left(a, b, \frac{2\eta x}{\sigma^2}\right)}{\partial \frac{2\eta x}{\sigma^2}}\frac{\partial \frac{2\eta x}{\sigma^2}}{\partial x} \\
&\quad -x^{2a}U\left(a, b, \frac{2\eta x}{\sigma^2}\right)\frac{\partial M\left(a, b, \frac{2\eta x}{\sigma^2}\right)}{\partial \frac{2\eta x}{\sigma^2}}\frac{\partial \frac{2\eta x}{\sigma^2}}{\partial x} \\
&= x^{2a}\frac{2\eta}{\sigma^2}W\left(y_1\left(\frac{2\eta x}{\sigma^2}\right), y_5\left(\frac{2\eta x}{\sigma^2}\right)\right).
\end{aligned}$$

Using (6.3),

$$W(\psi_1, \psi_4) = -\frac{\Gamma(b)}{\Gamma(a)}x^{2a-b}\exp\left(\frac{2\eta x}{\sigma^2}\right)\left(\frac{2\eta}{\sigma^2}\right)^{1-b}.$$

□

Lemma B.17.

$$W(\psi_2, \psi_4) = -\frac{\Gamma(2-b)}{\Gamma(1+a-b)}x^{2a-b}\exp\left(\frac{2\eta x}{\sigma^2}\right).$$

Proof. Using Lemma B.14,

$$\begin{aligned}
W(\psi_2,\psi_4) &= x^{2a}W\left(M\left(a,b,\frac{2\eta x}{\sigma^2}\right),x^{1-b}M\left(a+1-b,2-b,\frac{2\eta x}{\sigma^2}\right)\right)\\
&= x^{2a}x^{1-b}M\left(a+1-b,2-b,\frac{2\eta x}{\sigma^2}\right)\frac{\partial M\left(a,b,\frac{2\eta x}{\sigma^2}\right)}{\partial x}\\
&\quad -x^{2a}M\left(a,b,\frac{2\eta x}{\sigma^2}\right)\frac{\partial x^{1-b}M\left(a+1-b,2-b,\frac{2\eta x}{\sigma^2}\right)}{\partial x}\\
&= x^{2a}\left(\frac{2\eta}{\sigma^2}\right)^{b-1}x^{1-b}M\left(a+1-b,2-b,\frac{2\eta x}{\sigma^2}\right)\frac{\partial M\left(a,b,\frac{2\eta x}{\sigma^2}\right)}{\partial\frac{2\eta x}{\sigma^2}}\frac{\partial\frac{2\eta x}{\sigma^2}}{\partial x}\\
&= -x^{2a}\left(\frac{2\eta}{\sigma^2}\right)^{b-1}M\left(a,b,\frac{2\eta x}{\sigma^2}\right)\frac{\partial\left(\frac{2\eta x}{\sigma^2}\right)^{1-b}M\left(a+1-b,2-b,\frac{2\eta x}{\sigma^2}\right)}{\partial\frac{2\eta x}{\sigma^2}}\frac{\partial\frac{2\eta x}{\sigma^2}}{\partial x}\\
&= x^{2a}\left(\frac{2\eta}{\sigma^2}\right)^{b}W\left(y_2\left(\frac{2\eta x}{\sigma^2}\right),y_5\left(\frac{2\eta x}{\sigma^2}\right)\right).
\end{aligned}$$

Using (6.4),

$$W(\psi_2,\psi_4)=-\frac{\Gamma(2-b)}{\Gamma(1+a-b)}x^{2a-b}\exp\left(\frac{2\eta x}{\sigma^2}\right).$$

□

Convergence and Divergence

Lemma B.18. *The series $x^cM(a,b,x)$ is convergent for every $x<\infty$.*

Proof. Using the ratio test (Theorem A.3),

$$\begin{aligned}
L &= \lim_{n\to\infty}\left|\frac{\left(\frac{2\eta}{\sigma^2}\right)^{n+1}\frac{(a)_{n+1}}{(b)_{n+1}}\frac{x^{c+n+1}}{(n+1)!}}{\left(\frac{2\eta}{\sigma^2}\right)^{n}\frac{(a)_n}{(b)_n}\frac{x^{c+n}}{(n)!}}\right|\\
&= \lim_{n\to\infty}\left|\frac{2\eta x}{\sigma^2}\frac{a+n}{b+n}\frac{1}{n+1}\right|\\
&= 0.
\end{aligned}$$

Since $L<1$ the series converges. □

Lemma B.19. *The series $x^cY(a,b,x)$ is convergent for every $x<\infty$.*

Proof. Using the ratio test (Theorem A.3),

$$\begin{aligned} L &= \lim_{n\to\infty} \left| \frac{\left(\frac{2\eta}{\sigma^2}\right)^{n+1} \frac{(a)_{n+1}}{(b)_{n+1}} \frac{x^{a+n+1}}{(a+1+n)(n+1)!}}{\left(\frac{2\eta}{\sigma^2}\right)^{n} \frac{(a)_{n}}{(b)_{n}} \frac{x^{a+n}}{(a+n)(n)!}} \right| \\ &= \lim_{n\to\infty} \left| \frac{2\eta x}{\sigma^2} \frac{a+n}{b+n} \frac{a+n}{a+n+1} \frac{1}{n+1} \right| \\ &= 0. \end{aligned}$$

Since $L < 1$ the series converges. □

Lemma B.20.

$$\lim_{x\to\infty} \frac{\psi_1(x)}{x^{2a-b} \exp\left(\frac{2\eta x}{\sigma^2}\right)} = \left(\frac{2\eta}{\sigma^2}\right)^{a-b} \frac{\Gamma(b)}{\Gamma(a)}.$$

Proof. Using (6.5),

$$\begin{aligned} \lim_{x\to\infty} \frac{\psi_1(x)}{x^{2a-b} \exp\left(\frac{2\eta x}{\sigma^2}\right)} &= \lim_{x\to\infty} \frac{x^a M\left(a, b, \frac{2\eta x}{\sigma^2}\right)}{x^{2a-b} \exp\left(\frac{2\eta x}{\sigma^2}\right)} \\ &= \lim_{x\to\infty} \frac{x^a \frac{\Gamma(b)}{\Gamma(a)} \left(\frac{2\eta x}{\sigma^2}\right)^{a-b} \exp\left(\frac{2\eta x}{\sigma^2}\right) \left[1 + O\left(\left|\frac{2\eta x}{\sigma^2}\right|^{-1}\right)\right]}{x^{2a-b} \exp\left(\frac{2\eta x}{\sigma^2}\right)} \\ &= \left(\frac{2\eta}{\sigma^2}\right)^{a-b} \frac{\Gamma(b)}{\Gamma(a)}. \end{aligned}$$

□

Lemma B.21.

$$\lim_{x\to\infty} \frac{\psi_2(x)}{x^{2a-b} \exp\left(\frac{2\eta x}{\sigma^2}\right)} = \left(\frac{2\eta}{\sigma^2}\right)^{a-1} \frac{\Gamma(2-b)}{\Gamma(1+a-b)}.$$

Proof. Using (6.5),

$$\begin{aligned} \lim_{x\to\infty} \frac{\psi_2(x)}{x^{2a-b} \exp\left(\frac{2\eta x}{\sigma^2}\right)} &= \lim_{x\to\infty} \frac{x^{a+1-b} M(1+a-b, 2-b, z)}{x^{2a-b} \exp\left(\frac{2\eta x}{\sigma^2}\right)} \\ &= \lim_{x\to\infty} \frac{x^{a+1-b} \frac{\Gamma(2-b)}{\Gamma(1+a-b)} \left(\frac{2\eta x}{\sigma^2}\right)^{a-1} \exp\left(\frac{2\eta x}{\sigma^2}\right) \left[1 + O\left(\left|\frac{2\eta x}{\sigma^2}\right|^{-1}\right)\right]}{x^{2a-b} \exp\left(\frac{2\eta x}{\sigma^2}\right)} \\ &= \left(\frac{2\eta}{\sigma^2}\right)^{a-1} \frac{\Gamma(2-b)}{\Gamma(1+a-b)}. \end{aligned}$$

□

Lemma B.22.

$$\lim_{x\to\infty} \psi_4(x) = \left(\frac{2\eta}{\sigma^2}\right)^{-a}.$$

Proof. Using (6.6),

$$\begin{aligned}\lim_{x\to\infty} \psi_4(x) &= \lim_{x\to\infty} x^a \left(\frac{2\eta x}{\sigma^2}\right)^{-a} \left[1 + O\left(\left|\frac{2\eta x}{\sigma^2}\right|^{-1}\right)\right] \\ &= \left(\frac{2\eta}{\sigma^2}\right)^{-a}.\end{aligned}$$

□

Appendix C

Java Classes

The Java programming language was used to implement the fixed cost models in Chapter 3 and the decreasing cost models and error models in Chapter 4. It was also used to support the increasing capacity models in Chapters 5 and 6. The object-orientated capabilities of Java were used to define three abstract classes: **RandomVariates**, **StochasticProcess** and **Perpetual**. These were subsequently used to define

- one random variate: a normal variate,
- three stochastic processes:a geometric Brownian motion (GBM), a jump-diffusion process (JDP), and a logistic process (LP),
- two fixed-cost models: a GBM model and a JDP model.

In computer science, this concept is referred to as polymorphism. The polymorphic approach offers several advantages. The main advantage is that software is written for an abstract class does not need to duplicated for each separate case. This saves time and space and also eliminates the the need for re-testing.

In this thesis, the polymorphic approach enabled us to define

- one decreasing cost model,

- one cost error model, and
- one traffic error model,

which can be used by GBM and JDP models. We could have defined one class for the jump-diffusion model and then turned off the jumps (i.e. set $\phi = 0$) to obtain a geometric Brownian motion model. However, the current approach enables us to define additional GBM functions which exploit (2.2) and (2.5). We can also use the results from the simpler GBM code to validate those from the more complex JDP code. A further advantage is that the code will also support additional fixed-cost models which may be defined in the future.

C.1 Utility Classes

The utility classes provide utility functions for the other programs. The **Data** class performs basic statistical analysis. The **CumulativeNormalDistribution** class calculates the cumulative normal distribution using the polynomial approximation given in Theorem A.10. The **RandomVariates** class is an abstract class for generating random variates. The **NormalVariates** class generates a normal variate using the polar method given in Algorithm A.11.

Data.java

```
public class Data
    {
    public Data() {
        points = new double[10000];
        numPoints = 0;
        sigma1 = sigma2 = 0;
    }
```

```
public Data(int maxNumPoints) {
    points = new double[maxNumPoints];
    numPoints = 0;
    sigma1 = sigma2 = 0;
}

public void addPoint(double point){
    points[numPoints++] = point;
    sigma1 += point;
    sigma2 += point * point;
}

public void deleteAllPoints() {
    numPoints = 0;
    sigma1 = sigma2 = 0;
}

public double mean(){
    if (numPoints < 1) return 0;
    else return sigma1/numPoints;
}
public double variance(){
    if (numPoints < 2) return 0;
    else
    return (numPoints/(numPoints-1.0))
        *(sigma2/numPoints -(sigma1/numPoints)*(sigma1/numPoints));
}
```

```
    private double[] points;
    private int numPoints;
    private double sigma1, sigma2;
    private boolean ready;
}
```

CumulativeNormalDistribution.java

```
public class CumulativeNormalDistribution
{
    CumulativeNormalDistribution() {}
    public double value(double x) {
    double ax = Math.abs(x);
    double zx = Math.exp(- ax * ax /2.0)/sqrt_2_pi;
    double t = 1.0/(1.0 + p*ax);
    double t2 = t * t;
    double px = 1.0 - zx * t * (b1 + b2*t + b3*t2 + b4*t2*t +
                b5*t2*t2);
    if (x < 0.0)
        px = 1.0 - px;
    return (double) (px);

    }

    private static final double sqrt_2_pi = 2.506628275;
    private static final double p = .2316419;
    private static final double b1 = .319381530;
    private static final double b2 = -.356563782;
    private static final double b3 = 1.781477937;
```

```
    private static final double b4 = -1.821255978;
    private static final double b5 = 1.330274429;
}
```

RandomVariates.java

```
import java.util.Random;
public abstract class RandomVariates
    {
    public RandomVariates() {
        generator = new Random();
    }

    public abstract void printLabel();
    public abstract double gen();
    public abstract double mean();
    public abstract double variance();

    public void setSeed(long seed){
        generator.setSeed(seed);
    }

    protected Random generator;
}
```

NormalVariates.java

```
import java.io.*;
public class NormalVariates extends RandomVariates
    {
```

```
public NormalVariates() {
    super();
    mu = 0;
    sigma = 1;
    ready = false;
}

public NormalVariates(double _mu, double _sigma) {
    super();
    mu = _mu;
    sigma = _sigma;
    ready = false;
}

public void printLabel(){
    System.out.println("N("+ mu + "," + sigma + ")");
}

public double mean(){
    return mu;
}
public double variance(){
    return sigma*sigma;
}

public double gen(){
    if (ready) {
    ready = false;
    return mu + sigma * value;
```

```
        }
        else {
        // use polar method - see Law & Kelton pg 490
        double V1, V2, W, Y, X1, X2;
        do{
            V1 = 2*generator.nextDouble()-1;
            V2 = 2*generator.nextDouble()-1;
            W = Math.pow(V1,2) + Math.pow(V2,2);
        } while (W>1);
        Y = Math.sqrt((-2*Math.log(W))/W);
        X1 = V1*Y;
        X2 = V2*Y;
        value = X2;
        ready = true;
        return mu + sigma * X1;
        }
    }

    private double mu;
    private double sigma;
    private boolean ready;
    private double value;
}
```

C.2 Stochastic Process Classes

The following classes simulate stochastic processes. The **StochasticProcess** class is an abstract class for stochastic processes. The **GeometricBrownianMotion**, **JumpDiffusionProcess** and **LogisticProcess** classes implement the geometric

Brownian motion, jump-diffusion process and logistic process respectively.

StochasticProcess.java

```
import java.text.*;
import java.io.*;
public abstract class StochasticProcess
    {
    public StochasticProcess() {
        printStep = 1;
    }
    public abstract void setSeed(long seed);
    public abstract void printLabel();
    public abstract void reset();
    public abstract void step();
    public abstract double value();
    protected abstract double trend(double elapsedTime);
    protected abstract String dir();

    public void setPrintStep(int _printStep) {
        printStep = _printStep;
    }

    public boolean isTime(double time) {
        return (Math.abs(t - time) < dt/2);
    }

    public double getTime() {
        return t;
```

```
}

public void adjustValue(double offset) {
    x += offset;
}

public void walk(){
    reset();
    for (double t = t0; t < T; t += dt) {
    System.out.println(df.format(t) + ":"+ df.format(value()));
    step();
    }
    System.out.println(df.format(T) + ":"+ df.format(value()));
}

public Data run(int N){
    Data results = new Data(N);
    for (int n = 0; n < N; n++) {
    reset();
    for (double t = 0; t < T; t += dt) {
        step();
    }
    results.addPoint(value());
    }
    return results;
}

public String estimateStoppingTime(long seed, int N, double vstar){
    String str = new String("");
```

```
boolean exercise;
int stepCount;
double v = x0, g, f;
double tstar = -1, count = 0,  totTstar = 0,  totVT = 0;
double count1 = 0,  totTstar1 = 0,  totVT1 = 0;
Data ed = new Data(N);
Data pd = new Data(N);
Data ed1 = new Data(N);
Data pd1 = new Data(N);

if ((vstar < 0) || ((Double.isInfinite(vstar)))) {
        if (N < 2){
        str = str.concat(new String (df.format(tstar) + " "));
        }
        else {
        str = str.concat(new String (df.format(0) + " "));
        str = str.concat(new String (df.format(0)+ " "));
        str = str.concat(new String (df.format(0)+ " "));
        str = str.concat(new String (df.format(0)+ " "));
        str = str.concat(new String (df.format(-2)+ " "));
        str = str.concat(new String (df.format(0)+ " "));
        }
        return str;
}


try {
if (seed > 0) setSeed(seed);
for (int n = 0; n < N; n++) {
```

```
tstar = -1;
reset();
exercise = false;
stepCount = 0;
for (double t = t0; t < T; t += dt) {
step();
v = value();
if (v >= vstar) {
    if (!exercise) {
     tstar = t;
     exercise = true;
    }
}

if (Math.abs (t-(T/10.0)) < 0.5*dt ){
 if (tstar >= 0) {
    totTstar1 += tstar;
    ed1.addPoint(tstar);
    pd1.addPoint(1);
    count1++;
    }
    else {
    pd1.addPoint(0);
    }
    totVT1 += v;
}
}
```

```
        if (tstar >= 0) {
        totTstar += tstar;
        ed.addPoint(tstar);
        pd.addPoint(1);
        count++;
        }
        else {
        pd.addPoint(0);
        }
        totVT += v;
    }

    if (N < 2){
        if (tstar < 0) {

        str = str.concat(new String (df.format(tstar) + " "));

        } else {
        str = str.concat(new String (df.format(tstar)+ " "));

        }
    }
    else {
    str = str.concat(new String (df.format(count/N) + " "));
    str = str.concat(new String (df.format(totTstar/count)+ " "));
    str = str.concat(new String (df.format(pd.mean())+ " "));
    str = str.concat(new String (df.format(pd.variance())+ " "));
    str = str.concat(new String (df.format(ed.mean())+ " "));
    str = str.concat(new String (df.format(ed.variance())+ " "));
```

```
        str = str.concat(new String (df.format(count1/N) + " "));
        str = str.concat(new String (df.format(totTstar1/count1)+ " "));
        str = str.concat(new String (df.format(pd1.mean())+ " "));
        str = str.concat(new String (df.format(pd1.variance())+ " "));
        str = str.concat(new String (df.format(ed1.mean())+ " "));
        str = str.concat(new String (df.format(ed1.variance())+ " "));
        }
        } catch (Exception e) {
        System.out.println(e.toString());
        }
        return str;
    }

    public void samplePath(String filename){
        String filenames[] = new String[1];
        filenames[0] = filename;
        samplePaths(filenames);
    }

    public void samplePaths(){
        String filenames[] = new String[3];
        for (int i = 0; i < 3; i++){
        filenames[i] = "path" + String.valueOf(i+1) + ".out";
        }
        samplePaths(filenames);
    }

    protected void samplePaths(String filenames[]){
        int stepCount;
```

```
int dim = filenames.length;
File[] pF = new File[dim];
for (int i = 0; i < dim; i++) {
System.out.println(filenames[i]);
pF[i] = new File("C://"+dir() + filenames[i]);
}

File tF = new File("C://"+dir() + "trend.out");
FileWriter[] pOut = new FileWriter[dim];
FileWriter tOut;

try {
for (int i = 0; i < dim; i++) {
    pOut[i] = new FileWriter(pF[i]);
}
tOut = new FileWriter(tF);

for (int i = 0; i < dim; i++) {
    reset();
    stepCount = 0;
    for (double t = t0; ; t += dt) {
    if ((stepCount % printStep) == 0) {
        pOut[i].write(df.format(t) + " "
              + df.format(value()) + "\n");
        if (i == 0) {
        tOut.write(df.format(t) + " "
               + df.format(trend(t-t0)) + "\n");
        }
    }
```

```
            if (Math.abs(t-T) < dt/2) break;
            step();
            stepCount++;
            }
        }
        for (int i = 0; i < dim; i++) {
            pOut[i].close();
        }
        tOut.close();
        } catch (Exception e) {
        System.out.println(e.toString());
        }
    }

    protected DecimalFormat df = new DecimalFormat("##.####");
    protected double x;
    protected double x0;
    protected double t;
    protected double t0;
    protected double dt;
    protected double T;
    protected int printStep;
}
```

GeometricBrownianMotion.java

```
import java.util.Random;
import java.text.*;
import java.io.*;
```

```
public class GeometricBrownianMotion extends BrownianMotion
    {
    public GeometricBrownianMotion(double _x0, double _mu,
    double _sigma, double _t0, double _dt, double _T) {
        super(_x0,_mu,_sigma,_t0,_dt,_T);
    }
    public GeometricBrownianMotion( double _mu, double _sigma,
                      double _dt, double _T) {
        super(_mu,_sigma,_dt,_T);
    }

    public void printLabel(){
        System.out.println("Geometric Brownian Motion");
    }

    public void reset(){
        t= t0;
        x = x0;
    }

    public void step(){
        t += dt;
        if (exact){
            x *= Math.exp((mu - 0.5 *sigma* sigma)*dt +
            sigma*Math.sqrt(dt)*nvs.gen());
        }
        else{
            x *= (1 + mu*dt + sigma*Math.sqrt(dt)*nvs.gen());
```

```
        }
    }

    protected double trend(double elapsedTime) {
        return x0 *Math.pow(1+ mu *dt,elapsedTime/dt) ;
    }

    protected String dir() {
        return new String("gbm_");
    }

    protected boolean exact = false;
}
```

JumpDiffusionProcess.java

```
import java.util.Random;
import java.io.*;
public class JumpDiffusionProcess extends StochasticProcess
    {
    public JumpDiffusionProcess(double _x0, double _mu,
    double _sigma, double _lambda, double _p, double _eta1,
    double _eta2, double _t0, double _dt, double _T) {
        super();
        x0 = _x0;
        mu = _mu;
        sigma = _sigma;
        lambda  = _lambda;
        p = _p;
```

```
        eta1 = _eta1;
        eta2 = _eta2;
        t0 = _t0;
        dt = _dt;
        T = _T;
        nvs = new NormalVariates();
        generator = new Random();
        reset();
    }

    public JumpDiffusionProcess(double _mu, double _sigma,
    double _lambda, double _p, double _eta1, double _eta2,
    double _dt, double _T) {
        super();
        x0 = 0.0;
        mu = _mu;
        sigma = _sigma;
        lambda  = _lambda;
        p = _p;
        eta1 = _eta1;
        eta2 = _eta2;
        t0 = 0.0;
        dt = _dt;
        T = _T;
        nvs = new NormalVariates();
        generator = new Random();
        reset();
    }
```

```
public void printLabel(){
    System.out.println("Jump Diffusion Process");
}

public void reset(){
    x =x0;
}
public double value(){
    return x;
}

public void step(){
    double jump = 0;
    double jump_correction = - lambda * ( p * eta1/(eta1 - 1) +
                      (1-p) * eta2/(eta2+1) - 1);
    if (generator.nextDouble() <= (lambda * dt)) {
    if (generator.nextDouble() <= p)
        jump = -Math.log(generator.nextDouble())/eta1;
    else
        jump = Math.log(generator.nextDouble())/eta2;
    }

    x *= Math.exp((mu - 0.5 *sigma* sigma + jump_correction)*dt
        + sigma*Math.sqrt(dt)*nvs.gen() + jump);
}

public void setSeed(long seed){
    nvs.setSeed(seed);
    generator.setSeed(seed);
```

```
    }

    protected double trend(double elapsedTime) {
        double jump_correction = - lambda * ( p * eta1/(eta1 - 1) +
                                 (1-p) * eta2/(eta2+1) - 1);
        jump_correction = 0;
        return x0*Math.pow(1+(mu+jump_correction)*dt,elapsedTime/dt);
    }

    protected String dir() {
        return new String("jdp/");
    }

    protected NormalVariates nvs;
    protected Random generator;
    protected double mu;
    protected double sigma;
    protected double lambda;
    protected double p;
    protected double eta1;
    protected double eta2;

}
```

LogisticProcess.java

```
import java.util.Random;
import java.io.*;
public class LogisticProcess extends StochasticProcess
```

```
{
public LogisticProcess(double _x0, double _eta, double _xbar,
               double _sigma, double _t0, double _dt,
               double _T) {
    super();
    x0 = _x0;
    eta = _eta;
    xbar = _xbar;
    sigma = _sigma;
    t0 = _t0;
    dt = _dt;
    T = _T;
    nvs = new NormalVariates();
    reset();
}

public LogisticProcess(double _eta, double _xbar, double _sigma,
               double _dt, double _T) {
    super();
    x0 = 0.0;
    eta = _eta;
    xbar = _xbar;
    sigma = _sigma;
    t0 = 0.0;
    dt = _dt;
    T = _T;
    nvs = new NormalVariates();
    reset();
}
```

```
public void printLabel(){
    System.out.println("Logistic Process");
}


public void reset(){
    t = t0;
    x =x0;
}
public double value(){
    return x;
}
public void step(){
    t += dt;
    x = x + eta*(xbar-x)*x*dt + sigma*x*Math.sqrt(dt)*nvs.gen();
}


public void setSeed(long seed){
    nvs.setSeed(seed);
}


protected double trend(double elapsedTime) {
    return xbar;
}


protected String dir() {
    return new String("lp_");
}
```

```
    protected NormalVariates nvs;
    protected double eta;
    protected double xbar;
    protected double sigma;
}
```

C.3 Finite-Time Model Class

The **FiniteModel** class implements the finite-time models in Sections 3.4 and 4.2.5.

FiniteModel.java

```
import java.text.*;
import java.util.Random;
import java.io.*;
public class FiniteModel {

    public FiniteModel(double T, double s0, double k, double r,
    double delta, double alpha, double sigma)
    {
        this.t0 = 0;
        this.T = T;
        this.s0 = s0;
        this.k = k;
        this.r = r;
        this.delta = delta;
        this.alpha = alpha;
        this.sigma = sigma;
    }
```

```
public double value(int N){
    double dt = T/N;
    double u = Math.exp(sigma*Math.sqrt(dt));
    double d = Math.exp(-sigma*Math.sqrt(dt));
    double a = Math.exp((r-delta)*dt);
    double R = Math.exp(r*dt);
    double pi = (a-d)/(u-d);
    double [][] S = new double[N+1][N+1];
    double [][] E = new double[N+1][N+1];
    double [][] V = new double[N+1][N+1];
    double w;
    int [] st = new int[N+1];
    for (int j =0; j <= N; j++){
    st[j] = N;
    }

    for(int n =N; n >= 0; n--){
        double cost = k * Math.exp(-alpha*n*T/N);
    for (int j = 0; j <= n; j++) {
        S[n][j] = s0*Math.pow(u,j)*Math.pow(d,n-j);

        if (call)  {
        E[n][j] = Math.max(S[n][j] - cost,0);
        }
        else {
        E[n][j] = Math.max(cost - S[n][j],0);
        }
        if (n == N) {
        V[n][j] = E[n][j];
```

```
            }
            else {
                if(european) {
                    V[n][j] = (pi*V[n+1][j+1] + (1-pi)*V[n+1][j])/R;
                }
                else {

                    V[n][j] = Math.max((pi*V[n+1][j+1] +
                        (1-pi)*V[n+1][j])/R,E[n][j]);
                     }
            }
        }

        }

        return V[0][0];

    }
    protected double s0;
    protected double k;
    protected double delta;
    protected double r;
    protected double alpha;
    protected double sigma;
    protected double t0;
    protected double T;
    protected boolean european = false;
    protected boolean call = true;
}
```

C.4 Fixed-Cost Model Classes

The following classes implement the fixed-cost models in Chapter 3. The **Perpetual** class is an abstract class for fixed-cost models. The **PerpetualGBM** and **PerpetualJDP** classes implement the GBM model and JDP model respectively.

Perpetual.java

```
import java.text.*;
import java.io.*;
import java.util.*;
public abstract class Perpetual
    {
    public Perpetual() {
        calculated =  bisection = false;
        setDefaults();
        pathNo = 0;
        t0 = 0;
        dt = 0.01;
        T=100;
    }

    public Perpetual(String inputFile) {
        calculated = bisection = false;
        setDefaults();
        readInputs(inputFile);
        setOtherParameters();
        pathNo = 0;
        t0 = 0;
        dt = 0.01;
```

```
        T=100;
    }

    public Perpetual(Perpetual model) {
        calculated =  bisection = false;
        setDefaults();
        r = model.r;
        nu = model.nu;
        sigma = model.sigma;
        I = model.I;
        v0 = model.v0;
        pathNo = 0;
        t0 = 0;
        dt = 0.01;
        T=100;
    }

    public  abstract Perpetual copy();

    protected void printLabel(){
        System.out.println(label);
    }

    protected abstract void setDefaults();

    public abstract double root(double nu, double r);

    protected void setOtherParameters(){}
```

```
protected abstract void clearInputs();

public void setParameter(String param, double value){
    if (param.equals("r")) {
        r = value;
    }else if (param.equals("nu")){
        nu = value;
    }else if (param.equals("delta")){
        nu = r-value;
    }else   {
        System.out.println("System Error has occurred");
        System.exit(1);
    }
}

public double getParameter(String param){
    if (param.equals("r")) {
        return r;
    }else if (param.equals("nu")){
        return nu;
    }else   {
        System.out.println("System Error has occurred");
        return Double.NaN;
    }
}

protected abstract boolean insufficientInput();

protected abstract void calculateValues();
```

```
public abstract double value();

public abstract double trigger();

protected abstract String values();

protected abstract void sensAnalysis(String param, double start,
                          double end, double step, String filename);

public abstract String simulate(long seed, int N);

protected boolean investImmediately() {return false;}

protected boolean stochastic() {return false;}

protected double drift() {return 0;}

protected double mu() {return 0;}

protected double m(double x) {return 1;}

protected double P(){
    if (beta <= 1.00001) return 0;
    else if (investImmediately()) return 1;
    else if (stochastic()) {
        double m = m(vstar);
        double mu = mu();
        return Math.exp(m*mu-m*Math.abs(mu));
```

```
        }
        else if (drift() > 0) return 1;
        else return 0;
    }

    protected double P(double T){
        if (beta <= 1.00001) return 0;
        double m = m(vstar);
        double mu = mu();
        CumulativeNormalDistribution cnd =
            new CumulativeNormalDistribution();
        return cnd.value((-m+mu*T)/Math.sqrt(T))+
        Math.exp(2*m*mu)*cnd.value((-m-mu*T)/Math.sqrt(T));
    }



    public double expStopTime(){
        double m = m(vstar);
        double mu = mu();
        if (beta <= 1.00001) return -2;
        if (investImmediately()) return 0;
        else if (mu <= 0) return Double.POSITIVE_INFINITY;
        else return m/mu;
    }

    public double avgStopTime(double x){
        double m = m(x);
        double mu = mu();
        if ((beta <= 1.00001) | (x <= 0)) return -2;
```

```
        if (investImmediately()) return 0;
        else if (mu <= 0) return Double.POSITIVE_INFINITY;
        else return m/mu;
    }

    /** This function uses Simpson's rule */
    protected double expStopTime(double T){
        if (beta <= 1.00001) return 0;
        int  N = 5000;
        double I = 0;
        double h = (T-0.0)/(2*N);
        for(int i = 0; i <= 2*N; i++) {
        if ((i == 0) | (i == (2*N))) {
            I += P(i*h);
        }
        else if ((i % 2) == 0) {
            I += 2*P(i*h);
        }
        else {
            I += 4*P(i*h);
        }
        }

        I *= h/3.0;
        return (T*P(T) - I)/P(T);
    }

    public String stoppingTimes(){
        String str = new String("");
```

```
        calculateValues();
        str = str.concat(new String (df.format(T)+ " "));
        str = str.concat(new String (df.format(P())+ " "));
        str = str.concat(new String (df.format(P(T))+ " "));
        double tmp = expStopTime();
        if (tmp == Double.POSITIVE_INFINITY) tmp = -1 ;
        str = str.concat(new String (df.format(tmp)+ " "));
        str = str.concat(new String (df.format(expStopTime(T))+ " "));
        return str;
        }


    protected void readInputs(String inputFile){
        clearInputs();
        try {
        BufferedReader in =
            new BufferedReader(new FileReader(inputFile));
        while (in.ready()) {
            StringTokenizer st =
            new StringTokenizer(in.readLine(),"= ");
            if (st.hasMoreTokens()) {
            String param = st.nextToken().toLowerCase();
            double value = Double.parseDouble(st.nextToken());
            setParameter(param,value);
            }
        }

        if (insufficientInput()) {
            System.out.println("Error: Insufficient Input");
```

```
            System.exit(1);
        }

        } catch (Exception e) {
        System.out.println(e.toString());
        }

    }

    // input parameters
    protected double r;
    protected double nu;
    protected double sigma;
    protected double I;
    protected double v0;
    protected double t0;
    protected double T;
    protected double dt;
    protected double phi;
    protected double lambda;


    // output parameters
    protected int pathNo;
    protected double beta;
    protected double vstar;
    protected double fv;
    protected DecimalFormat df = new DecimalFormat("##.######");
    protected boolean calculated;
```

```
    protected boolean bisection;
    protected String label;
    protected String directory;
}
```

PerpetualGBM.java

```
import java.text.*;
import java.io.*;
import java.util.*;
public class PerpetualGBM extends Perpetual
    {
    public PerpetualGBM(double _r, double _nu, double _sigma,
                double _I, double _v0) {
        super();
        r = _r;
        nu = _nu;
        sigma = _sigma;
        I = _I;
        v0 = _v0;
    }

    public PerpetualGBM(String inputFile) {
        super(inputFile);
    }

    public PerpetualGBM(PerpetualGBM model) {
        super();
    }
```

```
    public  Perpetual copy(){
        Perpetual p = new PerpetualGBM(r,nu,sigma,I,v0);
        return p;
    }

    protected void setDefaults(){
        pathNo = 0;
        label = new String("Perpetual GBM");
        directory = new String("pgbm/");
    }

    protected void calculateValues() {
        if (!calculated){
        if ((sigma != 0)|(nu!=0)){
            beta = root(nu,r);
            vstar = VStar();
            if (!Double.isInfinite(vstar)){
                a = A();
                fv = F(v0);
                if (beta <= 1) fv = 0;
            } else {
                vstar = Double.MAX_VALUE;
                a = 0;
                fv = 0;
            }
        } else {
            vstar = I;
            fv = v0 - I;
```

```
        }
        p = P();
        calculated = true;
        }
    }

    protected void clearInputs() {
        r = nu = sigma = I = v0 = Double.NaN;
    }

    protected boolean insufficientInput() {
        double total = r + nu + sigma + I + v0;
        return ((new Double(total)).isNaN());
    }

    public double root(double nu, double r){
        if (sigma == 0) return r/(nu);
        return 0.5 - (nu)/(sigma*sigma) +
        Math.sqrt((0.5 - (nu)/(sigma*sigma))*
              (0.5 - (nu)/(sigma*sigma))
              + 2*r/(sigma*sigma));
    }

    protected double VStar(){
        return (beta * I)/(beta -1);
    }

    protected double A() {
        return (vstar - I)/(Math.pow(vstar,beta));
```

```
}

protected double F(double v){
    return a * Math.pow(v,beta);
}

protected boolean investImmediately() {
    return (v0 >= vstar);
}

protected boolean stochastic() {
    return (sigma != 0);
}

protected double drift() {
    return nu - 0.5*sigma*sigma;
}

protected double mu() {
    return (nu)/sigma - 0.5*sigma;
}

protected double m(double x) {
    return  Math.log(x/v0)/sigma;
}

public double value(){
    calculateValues();
    return fv;
```

```
}

public double trigger(){
    calculateValues();
    return vstar;
}

public String values(){
    String str = new String("");
    calculateValues();
    str = str.concat(new String (df.format(beta) + " "));
    str = str.concat(new String (df.format(a) + " "));
    str = str.concat(new String (df.format(v0) + " "));
    str = str.concat(new String (df.format(vstar) + " "));
    str = str.concat(new String (df.format(fv)+ " "));
    str = str.concat(new String (df.format(T)+ " "));
    str = str.concat(new String (df.format(p)+ " "));
    str = str.concat(new String (df.format(P(T))+ " "));
    double tmp = expStopTime();
    if (tmp == Double.POSITIVE_INFINITY) tmp = -1 ;
    str = str.concat(new String (df.format(tmp)+ " "));
    str = str.concat(new String (
        df.format(expStopTime(T))+ " "));
    return str;
}

public double getParameter(String param){
   if (param.equals("sigma")){
    return sigma;
```

```
    }else if (param.equals("i")){
    return I;
    }else if (param.equals("v0")){
    return v0;
    }else {
        return super.getParameter(param);
    }
}

public void setParameter(String param, double value){
    if (param.equals("sigma")){
        sigma = value;
    }else if (param.equals("i")){
        I = value;
    }else if (param.equals("v0")){
        v0 = value;
    }else if (param.equals("delta")){
        nu = r-value;
    }else {
        super.setParameter(param,value);
    }
}

protected void sensAnalysis(String param, double start,
double end, double step, String filename){
    double origBeta, origVstar, origA,  origV0, origFv, origP;

    calculateValues();
    origBeta = beta;
```

```
origVstar = vstar;
origA = a;
origV0 = v0;
origFv = fv;
origP = p;
File vF = new File(filename);
FileWriter vOut;

try {
vOut = new FileWriter(vF);
double x = start, min = 1;
for (double val = start; val <= end; val += step) {
    setParameter(param,val);
    setOtherParameters();
    calculated = false;
    calculateValues();
    vOut.write(df.format(val) + " "
           + values() + "\n");
}
System.out.println("min is " + min + " at " + x);
vOut.close();
} catch (Exception e) {
System.out.println(e.toString());
}

beta = origBeta;
vstar = origVstar;
a = origA;
v0 = origV0;
```

```
        fv = origFv;
        p = origP;
    }

    protected void stoppingTimes(String param, double start,
    double end, double step, String filename){

        File vF = new File(filename);
        FileWriter vOut;

        try {
        vOut = new FileWriter(vF);
        double x = start, min = 1;
        for (double val = start; val <= end; val += step) {
            setParameter(param,val);
            calculated = false;
            vOut.write(df.format(val) + " "
                   + stoppingTimes() + "\n");

        }
        System.out.println("min is " + min + " at " + x);
        vOut.close();
        } catch (Exception e) {
        System.out.println(e.toString());
        }

}

    protected void estimateStoppingTimes(String param,
```

```
double start, double end, double step, String filename,
long seed, int N){
        String str = new String("");



        for (double val = start; val <= end; val += step){
            setParameter(param,val);
            calculated = false;
            str = str.concat(new String (df.format(val) + " "));
            str = str.concat(new String (simulate(seed,N) + "\n"));

        }

        File fF = new File(filename);
        FileWriter fOut;



        try {
        fOut = new FileWriter(fF);
        fOut.write(str);
        fOut.close();

        } catch (Exception e) {
        System.out.println(e.toString());

        }
}

public String simulate(long seed, int N){
```

```
String str = new String("");
boolean exercise;
int stepCount;
double v = v0, g, f;
double tstar = -1, count = 0, totTstar = 0, totVT = 0;
Data ed = new Data();
Data pd = new Data();
gbm = new GeometricBrownianMotion(v0, nu, sigma,t0, dt, T);
calculateValues();

if ((vstar < 0) || ((Double.isInfinite(vstar)))) {
        System.out.println(vstar);
        System.out.println(beta==1);
        if (N < 2){
        str = str.concat(new String (df.format(tstar) + " "));
        }
        else {
        str = str.concat(new String (df.format(0) + " "));
        str = str.concat(new String (df.format(0)+ " "));
        str = str.concat(new String (df.format(0)+ " "));
        str = str.concat(new String (df.format(0)+ " "));
        str = str.concat(new String (df.format(-2)+ " "));
        str = str.concat(new String (df.format(0)+ " "));
        }
        return str;
}
try {
if (seed > 0) gbm.setSeed(seed);
for (int n = 0; n < N; n++) {
```

```
        tstar = -1;
        gbm.reset();
        exercise = false;
        stepCount = 0;
        for (double t = t0; t < T; t += dt) {
        gbm.step();
        v = gbm.value();
        if (v >= vstar) {
            if (!exercise) {
             tstar = t;
             exercise = true;
            }
        }
        }
        if (tstar >= 0) {
        totTstar += tstar;
        ed.addPoint(tstar);
        pd.addPoint(1);
        count++;
        }
        else {
        pd.addPoint(0);
        }
        totVT += v;
    }

    if (N < 2){
        if (tstar < 0) {
```

```
            str = str.concat(new String (df.format(tstar) + " "));

            } else {
            str = str.concat(new String (df.format(tstar)+ " "));


            }


        }
        else {
        str = str.concat(new String (df.format(count/N) + " "));
        str = str.concat(new String (df.format(totTstar/count)+ " "));
        str = str.concat(new String (df.format(pd.mean())+ " "));
        str = str.concat(new String (df.format(pd.variance())+ " "));
        str = str.concat(new String (df.format(ed.mean())+ " "));
        str = str.concat(new String (df.format(ed.variance())+ " "));
        }
        } catch (Exception e) {
        System.out.println(e.toString());
        }
        return str;
    }

    // output parameters
    protected GeometricBrownianMotion gbm;
    protected double a;
    protected double p;
}
```

PerpetualJDP.java

```
import java.text.*;
import java.io.*;
import java.util.*;
public class PerpetualJDP extends Perpetual
    {

    public PerpetualJDP(double _alpha, double _sigma, double _phi,
               double _lambda, double _r, double _I, double _v0) {
        super();
        nu = _alpha;
        sigma = _sigma;
        phi = _phi;
        lambda = _lambda;
        r = _r;
        I = _I;
        v0 = _v0;
    }

    public PerpetualJDP(String inputFile) {
        super(inputFile);
    }

    public Perpetual copy(){
        Perpetual p = new PerpetualJDP(nu,sigma,phi,lambda,r,I,v0);
        return p;
    }
```

```
protected void setDefaults(){
    bisection = true;
    label = new String("Lassila Five");
    directory = new String("lf/");
}

protected void calculateValues() {
    if (!calculated){
    beta = root(nu,r);

    vstar = VStar();
    if (!Double.isInfinite(vstar)) {
    a = A();
    fv = FV();
    } else {
        vstar = 0;
        a = 0;
        fv = 0;
    }
    calculated = true;
    }
}

protected void clearInputs() {
    nu = sigma = phi = lambda = r = I  = v0 = Double.NaN;
}

protected boolean insufficientInput() {
    double total = sigma + nu + phi + lambda + r + I + v0;
```

```
        return ((new Double(total)).isNaN());
    }

    protected  double rootFunction(double beta, double nu, double r){
         return 0.5 * sigma * sigma * beta * (beta -1.0) + nu* beta
        - (r + lambda) + lambda * Math.pow((1 - phi),beta);
    }

    protected  double rootFunction1(double beta, double nu, double r){
         return sigma * sigma * beta  + (nu-0.5 * sigma * sigma)
         +Math.log(1-phi) * lambda * Math.pow((1 - phi),beta);
    }

    public double root(double nu, double r){
        double k = 100;
        for (int i = 0; i < 10; i++){
            k = k - rootFunction(k,nu,r)/rootFunction1(k,nu,r);
        }
        return k;
    }

    protected double VStar(){
        return (beta * I)/(beta -1);
    }

    protected double A() {
        return (vstar - I)/(Math.pow(vstar,beta));
    }
```

```
protected double FV(){
    if (beta < 1) return 0;
    return a * Math.pow(v0,beta);
}

public double value(){
    calculateValues();
    return fv;
}

public double trigger(){
    calculateValues();
    return vstar;
}

public String values(){
    String str = new String("");
    calculateValues();
    str = str.concat(new String (df.format(beta) + " "));
    str = str.concat(new String (df.format(vstar) + " "));
    str = str.concat(new String (df.format(fv)+ " "));
    return str;
}

public void setParameter(String param, double val){
    if (param.equals("nu")) {
        nu = val;
    }else if (param.equals("sigma")){
        sigma = val;
```

```
        }else if (param.equals("phi")){
            phi = val;
        }else if (param.equals("lambda")){
            lambda = val;
        }else if (param.equals("r")){
            r = val;
        }else if (param.equals("i")){
            I = val;
        }else if (param.equals("v0")){
            v0 = val;
        }else if (param.equals("delta")){
            nu = r-val;
        }else {
            super.setParameter(param,val);
        }
    }

    protected void sensAnalysis(String param, double start,
    double end, double step, String filename){

        double origBeta, origVstar, origA, origFv;

        calculateValues();
        origBeta = beta;
        origVstar = vstar;
        origA = a;
        origFv = fv;
```

```
File vF = new File(filename);
FileWriter vOut;

try {
vOut = new FileWriter(vF);
for (double val = start; val <= end; val += step) {
    setParameter(param,val);
    beta = root(nu,r);
    if (beta != 1) {
        vstar = VStar();
        a = A();
        fv = FV();
    } else {
        vstar = Double.MAX_VALUE;
        a = 0;
        fv = 0;
    }
    if (vstar <= v0) fv = 0;
    vOut.write(df.format(val) + " "
           + values() + "\n");

}

vOut.close();
} catch (Exception e) {
System.out.println(e.toString());
}

beta = origBeta;
```

```
        vstar = origVstar;
        a = origA;
        fv = origFv;
    }

    protected void estimateStoppingTimes(String param,
    double start, double end, double step, String filename,
    long seed, int N){
            String str = new String("");



            for (double val = start; val <= end; val += step){
                setParameter(param,val);
                calculated = false;
                str = str.concat(new String (df.format(val) + " "));
                str = str.concat(new String (simulate(seed,N) + "\n"));

            }

            File fF = new File(filename);
            FileWriter fOut;



            try {
            fOut = new FileWriter(fF);
            fOut.write(str);
            fOut.close();

            } catch (Exception e) {
```

```
            System.out.println(e.toString());

            }
    }

    public String simulate(long seed, int N){
        String str = new String("");
        boolean exercise;
        int stepCount;
        double v = v0, g, f;
        double tstar = -1, count = 0,  totTstar = 0,  totVT = 0;
        double count1 = 0,  totTstar1 = 0,  totVT1 = 0;

        Data ed = new Data();
        Data pd = new Data();
        Data ed1 = new Data();
        Data pd1 = new Data();

        JumpDiffusionProcess jd =
        new JumpDiffusionProcess(v0, nu-lambda*phi, sigma, lambda,
                     0.0, 5000, (1-phi)/phi,t0, dt, T);
        calculateValues();

        if ((vstar < 0) || ((Double.isInfinite(vstar)))) {
                System.out.println(vstar);
                System.out.println(beta==1);
                if (N < 2){
                str = str.concat(new String (df.format(tstar) + " "));
                }
```

```
        else {
        str = str.concat(new String (df.format(0) + " "));
        str = str.concat(new String (df.format(0)+ " "));
        str = str.concat(new String (df.format(0)+ " "));
        str = str.concat(new String (df.format(0)+ " "));
        str = str.concat(new String (df.format(-2)+ " "));
        str = str.concat(new String (df.format(0)+ " "));
        }
        return str;
}


try {
if (seed > 0) jd.setSeed(seed);
for (int n = 0; n < N; n++) {
    tstar = -1;
    jd.reset();
    exercise = false;
    stepCount = 0;
    for (double t = t0; t < T; t += dt) {
    jd.step();
    v = jd.value();
    if (v >= vstar) {
        if (!exercise) {
         tstar = t;
         exercise = true;
        }
    }
```

```
            if (Math.abs (t-(T/10.0)) < 0.5*dt ){

             if (tstar >= 0) {
                totTstar1 += tstar;
                ed1.addPoint(tstar);
                pd1.addPoint(1);
                count1++;
                }
                else {
                pd1.addPoint(0);
                }
                totVT1 += v;
            }
            }

            if (tstar >= 0) {
            totTstar += tstar;
            ed.addPoint(tstar);
            pd.addPoint(1);
            count++;
            }
            else {
            pd.addPoint(0);
            }
            totVT += v;
        }

        if (N < 2){
            if (tstar < 0) {
```

```
          str = str.concat(new String (df.format(tstar) + " "));

          } else {
          str = str.concat(new String (df.format(tstar)+ " "));

          }

      }
      else {
      str = str.concat(new String (df.format(count/N) + " "));
      str = str.concat(new String (df.format(totTstar/count)+ " "));
      str = str.concat(new String (df.format(pd.mean())+ " "));
      str = str.concat(new String (df.format(pd.variance())+ " "));
      str = str.concat(new String (df.format(ed.mean())+ " "));
      str = str.concat(new String (df.format(ed.variance())+ " "));
      str = str.concat(new String (df.format(count1/N) + " "));
      str = str.concat(new String (df.format(totTstar1/count1)+ " "));
      str = str.concat(new String (df.format(pd1.mean())+ " "));
      str = str.concat(new String (df.format(pd1.variance())+ " "));
      str = str.concat(new String (df.format(ed1.mean())+ " "));
      str = str.concat(new String (df.format(ed1.variance())+ " "));
      }

      } catch (Exception e) {
      System.out.println(e.toString());
      }
      return str;
  }
```

```
    // output parameters
    protected double a;

}
```

C.5 Decreasing Cost and Error Model Classes

The following classes implement the decreasing cost and error models in Chapters 4. The **CostModel**, **CostErrorModel** and **TrafficErrorModel** classes implement the decreasing cost model, cost error model and traffic error model respectively.

CostModel.java

```
import java.text.*;
import java.io.*;
import java.util.*;
public  class CostModel
    {

    public CostModel(Perpetual _model) {
        model = _model;
        model.calculated = false;
        rate = model.r;
        drift = model.nu;
    }

    protected  String values(double alpha){
        String retStr = new String("");
```

```
        model.r = rate + alpha;
        model.nu = drift + alpha;
        model.calculated = false;
        retStr = model.values();
        return retStr;
    }

    protected void values(double start, double end,
        double step, String filename){
        File vF = new File(filename);
        FileWriter vOut;

        try {
        vOut = new FileWriter(vF);
        double x = start, min = 1;
        for (double val = start; val <= end; val += step) {
            double alpha = val;
            model.r = rate + alpha;
            model.nu = drift + alpha;
            model.calculated = false;
            vOut.write(df.format(val) + " "
                    + model.values() + "\n");
        }
        System.out.println("min is " + min + " at " + x);
        vOut.close();
        } catch (Exception e) {
        System.out.println(e.toString());
        }
    }
```

```
protected void stoppingTimes(double start, double end,
                double step, String filename){
    File vF = new File(filename);
    FileWriter vOut;
    try {
    vOut = new FileWriter(vF);
    double x = start, min = 1;
    for (double val = start; val <= end; val += step) {
        double alpha = val;
        model.r = rate + alpha;
        model.nu = drift + alpha;
        model.calculated = false;
        vOut.write(df.format(val) + " "
               + model.stoppingTimes() + "\n");
    }
    System.out.println("min is " + min + " at " + x);
    vOut.close();
    } catch (Exception e) {
    System.out.println(e.toString());
    }
}

protected void estimateStoppingTimes(double start, double end,
                double step, String filename, long seed, int N){
        String str = new String("");
        String mdstr = new String("");
        for (double alpha = start; alpha <= end; alpha += step){
            model.r = rate + alpha;
```

```
                model.nu = drift + alpha;
                model.calculated = false;
                str = str.concat(new String (df.format(alpha) + " "));
                mdstr = new String (model.simulate(seed,N) + "\n");
                str = str.concat(mdstr);
                System.out.print(model.r + " " + model.nu + mdstr);
            }

            File fF = new File(filename);
            FileWriter fOut;
            try {
            fOut = new FileWriter(fF);
            fOut.write(str);
            fOut.close();
            } catch (Exception e) {
            System.out.println(e.toString());
            }
    }

    // input parameters
    protected double rate;
    protected double drift;

    protected Perpetual model;
    protected DecimalFormat df = new DecimalFormat("##.######");
}
```

CostErrorModel.java

```
import java.text.*;
import java.io.*;
import java.util.*;
public  class CostErrorModel
{
    public CostErrorModel(Perpetual model) {
        rate = model.r;
        drift = model.nu;
        sigma = model.sigma;
        I = model.I;
        real = model;
        predicted = model.copy();
        predicted.calculateValues();
        System.out.println(predicted.v0);
    }

    protected String values(){
        String retstr = new String("");
        retstr = retstr.concat(df.format(alpha)+" ");
        real.r = rate + alpha;
        real.nu = drift + alpha;
        real.calculated = false;
        real.calculateValues();
        retstr = retstr.concat(df.format(alpha1)+" ");
        predicted.r = rate + alpha1;
        predicted.nu = drift + alpha1;
        predicted.calculated = false;
```

```
        predicted.calculateValues();
        retstr = retstr.concat(df.format(real.beta) + " ");
        retstr = retstr.concat(
            df.format(beta1 = predicted.beta) + " ");
        retstr = retstr.concat(df.format(beta2 =
            predicted.root(drift+alpha1,rate+alpha)) + " ");
        retstr = retstr.concat(df.format(beta1 - 1 - beta2) + " ");
        retstr = retstr.concat(df.format(real.fv) + " ");
        retstr = retstr.concat(df.format(predicted.fv) + " ");
        vstar1 = predicted.vstar;
        retstr = retstr.concat(df.format(G(real.v0)) + " ");
        double e1 = real.expStopTime();
        if (e1 == Double.POSITIVE_INFINITY) e1 = -1;
        retstr = retstr.concat(df.format(e1) + " ");
        double e2 = predicted.expStopTime();
        if (e2 == Double.POSITIVE_INFINITY) e2 = -1;
        retstr = retstr.concat(df.format(e2) + " ");
        retstr = retstr.concat(df.format(real.P()) + " ");
        retstr = retstr.concat(df.format(predicted.P()) + " ");
        retstr = retstr.concat("\n");
        return retstr;
    }

    protected String relerr(double i){
        String retstr = new String("");
        double disp = i*50;
        double pre, cor, act;
        retstr = retstr.concat(df.format(disp)+" ");
        retstr = retstr.concat(df.format(alpha1)+" ");
```

```
        real.r = rate + alpha;
        real.nu = drift + alpha;
        real.calculated = false;
        real.calculateValues();
        predicted.r = rate + alpha1;
        predicted.nu = drift + alpha1;
        predicted.calculated = false;
        predicted.calculateValues();
        retstr = retstr.concat(df.format(real.beta) + " ");
        retstr = retstr.concat(
            df.format(beta1 = predicted.beta) + " ");
        retstr = retstr.concat(df.format(beta2 =
            predicted.root(drift+alpha1,rate+alpha)) + " ");
        retstr = retstr.concat(df.format(beta1 - 1 - beta2) + " ");
        retstr = retstr.concat(df.format(cor = real.fv) + " ");
        retstr = retstr.concat(df.format(pre = predicted.fv) + " ");
        vstar1 = predicted.vstar;
        retstr = retstr.concat(df.format(act = G(real.v0)) + " ");
        retstr = retstr.concat(df.format((cor-pre)/cor) + " ");
        retstr = retstr.concat(df.format((cor-act)/cor) + " ");
        double e1 = real.expStopTime();
        if (e1 == Double.POSITIVE_INFINITY) e1 = -1;
        retstr = retstr.concat(df.format(e1) + " ");
        double e2 = predicted.expStopTime();
        if (e2 == Double.POSITIVE_INFINITY) e2 = -1;
        retstr = retstr.concat(df.format(e2) + " ");
        retstr = retstr.concat(df.format(real.P()) + " ");
        retstr = retstr.concat(df.format(predicted.P()) + " ");
        retstr = retstr.concat("\n");
```

```
        return retstr;
    }


    public double G(double v){
        return vstar1 * Math.pow(v/vstar1,beta1)
        - I * Math.pow(v/vstar1,beta2);
    }


    protected void varyGrowthParameter(int i, double start,
        double end, double step){
        String outdir = new String("C:/");
        File fF = new File(outdir + "err.out");
        FileWriter fOut;
        if (i==0)  return;
        if (i > 0) fF = new File(outdir + "err_opt"
            + (new Integer(i*50)).toString() +".out");
        if (i < 0) fF = new File(outdir + "err_pes"
            + (new Integer(-i*50)).toString() +".out");

        String outstr = new String("");
        for (double val = start; val <= end; val += step) {
            alpha = val;
            alpha1 = val*(1+0.5*i);
            if (i < 0 ) alpha1 = val/(1-0.5*i);
            outstr = outstr.concat(values());
        }
            try {
            fOut = new FileWriter(fF);
            fOut.write(outstr);
```

```
            fOut.close();
            } catch (Exception e) {
            System.out.println(e.toString());
            }
        }


    protected void varyRelativeError(double _alpha, double start,
        double end, double step){
        String outdir = new String("C:/");
        File fF = new File(outdir + "rel_err.out");
        FileWriter fOut;
        alpha = _alpha;
        String outstr = new String("");
        for (double i = start; i <= end; i += step) {
            alpha1 = alpha*(1+0.5*i);
            if (i < 0 ) alpha1 = alpha/(1-0.5*i);
            outstr = outstr.concat(relerr(i));
        }
            try {
            fOut = new FileWriter(fF);
            fOut.write(outstr);
            fOut.close();
            } catch (Exception e) {
            System.out.println(e.toString());
            }
    }

    // input parameters
```

```
    protected double rate;
    protected double drift;
    protected double sigma;
    protected double v0;
    protected double I;
    protected double alpha;
    protected double alpha1;
    // output parameters
    protected double beta1;
    protected double vstar1;
    protected double fv1;
    protected double beta2;
    protected double gv;
    protected Perpetual predicted;
    protected Perpetual real;
    protected DecimalFormat df = new DecimalFormat("##.####");
}
```

TrafficErrorModel.java

```
import java.text.*;
import java.io.*;
import java.util.*;
public  class TrafficErrorModel
    {
    public TrafficErrorModel(Perpetual model) {
        rate = model.r;
        drift = model.nu;
        sigma = model.sigma;
```

```
        I = model.I;
        real = model;
        predicted = model.copy();
        predicted.calculateValues();
        System.out.println(predicted.v0);
    }

    protected String values(){
        String retstr = new String("");
        real.r = rate;
        real.nu = drift + alpha;
        real.calculated = false;
        real.calculateValues();
        predicted.r = rate;
        predicted.nu = drift + alpha1;
        predicted.calculated = false;
        predicted.calculateValues();
        retstr = retstr.concat(df.format(real.beta) + " ");
        retstr = retstr.concat(df.format(beta1 = predicted.beta) + " ");
        retstr = retstr.concat(df.format(beta2 =
            predicted.root(drift+alpha1,rate+alpha1-alpha)) + " ");
        retstr = retstr.concat(df.format(beta1 - 1 - beta2) + " ");
        retstr = retstr.concat(df.format(real.vstar) + " ");
        retstr = retstr.concat(df.format(predicted.vstar) + " ");
        retstr = retstr.concat(df.format(real.fv) + " ");
        retstr = retstr.concat(df.format(predicted.fv) + " ");
        vstar1 = predicted.vstar;
        retstr = retstr.concat(df.format(G(real.v0)) + " ");
        double e1 = real.expStopTime();
```

```
        if (e1 == Double.POSITIVE_INFINITY) e1 = -1;
        retstr = retstr.concat(df.format(e1) + " ");
        double e2 = predicted.expStopTime();
        if (e2 == Double.POSITIVE_INFINITY) e2 = -1;
        retstr = retstr.concat(df.format(e2) + " ");
        double e3=  real.avgStopTime(predicted.vstar);
        if (e3 == Double.POSITIVE_INFINITY) e3 = -1;
        retstr = retstr.concat(df.format(e3) + " ");
        retstr = retstr.concat(df.format(real.P()) + " ");
        retstr = retstr.concat(df.format(predicted.P()) + " ");
        retstr = retstr.concat("\n");
        return retstr;
    }

    protected String relerr(double i){
        String retstr = new String("");
        double disp = i*50;
        double pre, cor, act;
        retstr = retstr.concat(df.format(disp)+" ");
        retstr = retstr.concat(df.format(alpha1)+" ");
        real.r = rate + alpha;
        real.nu = drift + alpha;
        real.calculated = false;
        real.calculateValues();
        predicted.r = rate + alpha1;
        predicted.nu = drift + alpha1;
        predicted.calculated = false;
        predicted.calculateValues();
        retstr = retstr.concat(df.format(real.beta) + " ");
```

```
        retstr = retstr.concat(df.format(beta1 =
        predicted.beta) + " ");
        retstr = retstr.concat(df.format(beta2 =
        predicted.root(drift+alpha1-alpha,rate+alpha1-alpha)) + " ");
        retstr = retstr.concat(df.format(beta1 - 1 - beta2) + " ");
        retstr = retstr.concat(df.format(cor = real.fv) + " ");
        retstr = retstr.concat(df.format(pre = predicted.fv) + " ");
        vstar1 = predicted.vstar;
        retstr = retstr.concat(df.format(act = G(real.v0)) + " ");
        retstr = retstr.concat(df.format((cor-pre)/cor) + " ");
        retstr = retstr.concat(df.format((cor-act)/cor) + " ");
        double e1 = real.expStopTime();
        if (e1 == Double.POSITIVE_INFINITY) e1 = -1;
        retstr = retstr.concat(df.format(e1) + " ");
        double e2 = predicted.expStopTime();
        if (e2 == Double.POSITIVE_INFINITY) e2 = -1;
        retstr = retstr.concat(df.format(e2) + " ");
        retstr = retstr.concat(df.format(real.P()) + " ");
        retstr = retstr.concat(df.format(predicted.P()) + " ");
        retstr = retstr.concat("\n");
        return retstr;
    }

    public double G(double v){
        if (beta1 <= 1) return 0;
        return vstar1 * Math.pow(v/vstar1,real.beta)
            - I * Math.pow(v/vstar1,real.beta);
    }
```

```
protected void varyGrowthParameter(int i, double start,
double end, double step){
    String outdir = new String("C:/prm_");
    File fF = new File(outdir + "errprm.out");
    FileWriter fOut;

    if (i==0)  return;
    if (i > 0) fF = new File(outdir + "err_opt"
        + (new Integer(i*50)).toString() +".out");
    if (i < 0) fF = new File(outdir + "err_pes"
        + (new Integer(-i*50)).toString() +".out");

    String outstr = new String("");
    for (double val = start; val <= end; val += step) {
        double nuD = val;
        double nuD1 = val*(1+0.5*i);
        if (i < 0 ) nuD1 = val/(1-0.5*i);
        outstr = outstr.concat(df.format(nuD)+ " ");
        outstr = outstr.concat(df.format(nuD1)+ " ");
        alpha = nuD;
        alpha1 = nuD1;
        outstr = outstr.concat(values());
    }
    try {
        fOut = new FileWriter(fF);
        fOut.write(outstr);
        fOut.close();
        } catch (Exception e) {
        System.out.println(e.toString());
```

```
        }
      }

  protected void varyRelativeError(double _alpha, double start,
  double end, double step){
      String outdir = new String("C:/");
      File fF = new File(outdir + "rel_err.out");
      FileWriter fOut;
      alpha = _alpha;
      String outstr = new String("");
      for (double i = start; i <= end; i += step) {
          alpha1 = alpha*(1+0.5*i);
          if (i < 0 ) alpha1 = alpha/(1-0.5*i);

          outstr = outstr.concat(relerr(i));
      }
      try {
          fOut = new FileWriter(fF);
          fOut.write(outstr);
          fOut.close();
          } catch (Exception e) {
          System.out.println(e.toString());

          }
  }

  protected void varyGrowthChange(int i, double start, double end,
      double step){
```

```
        String outdir = new String("C:/cng_");
        File fF = new File(outdir + "errchg.out");
        FileWriter fOut;
        if (i==0)  return;
        if (i > 0) fF = new File(outdir + "err_opt"
            + (new Integer(i*50)).toString() +".out");
        if (i < 0) fF = new File(outdir + "err_pes"
            + (new Integer(-i*50)).toString() +".out");
        String outstr = new String("");
        for (double val = start; val <= end; val += step) {
            double nuD = val;
            double nuD1 = val*(1+0.5*i);
            if (i < 0 ) nuD1 = val/(1-0.5*i);
            outstr = outstr.concat(df.format(nuD)+ " ");
            outstr = outstr.concat(df.format(nuD1)+ " ");
            alpha = 0;
            alpha1 = nuD1-nuD;
            outstr = outstr.concat(values());
        }
            try {
            fOut = new FileWriter(fF);
            fOut.write(outstr);
            fOut.close();
            } catch (Exception e) {
            System.out.println(e.toString());
            }
         }

    // input parameters
```

```
    protected double rate;
    protected double drift;
    protected double sigma;
    protected double v0;
    protected double I;
    protected double alpha;
    protected double alpha1;
    // output parameters
    protected double beta1;
    protected double vstar1;
    protected double fv1;
    protected double beta2;
    protected double gv;

    protected Perpetual predicted;
    protected Perpetual real;
    protected DecimalFormat df = new DecimalFormat("##.######");
}
```

C.6 Increasing Capacity Class

The MATLAB language was used to implement the increasing capacity models in Chapters 5 and 4 and the associated M-files are provided in Appendix D. The **IncreasingCapacity** class supports these models in the following ways:

- Calculates $C = E\left[\int_0^\infty \min(D(s), S)e^{-rs}ds\right]$ using Lemma 5.9.
- Estimates the stopping times and C-values in Chapter 6 using simulation techniques.

IncreasingCapacity.java

```
import java.text.*;
import java.io.*;
import java.util.*;
public class IncreasingCapacity
    {
    public IncreasingCapacity(String inputFile) {
         readInputs(inputFile);
         dt = (T-t0)/(2*M);
         if (dbar < 0) {
            sp = new GeometricBrownianMotion(d0,nu,sigma,t0,dt,T);
         }
         else{
            sp = new LogisticProcess(d0,nu/dbar,dbar,sigma,t0,dt,T);
         }
    }

    public IncreasingCapacity(StringTokenizer st) {
         readInputs(st);
         dt = (T-t0)/(2*M);
         if (dbar < 0) {
            sp = new GeometricBrownianMotion(d0,nu,sigma,t0,dt,T);
         }
         else{
            sp = new LogisticProcess(d0,nu/dbar,dbar,sigma,t0,dt,T);
         }
    }
```

```
protected void readInputs(String inputFile){
    try {
    BufferedReader in =
        new BufferedReader(new FileReader(inputFile));
        dbar = Double.parseDouble(in.readLine());
        nu = Double.parseDouble(in.readLine());
        sigma = Double.parseDouble(in.readLine());
        r = Double.parseDouble(in.readLine());
        beta = Double.parseDouble(in.readLine());
        alpha = Double.parseDouble(in.readLine());
        d0 = Double.parseDouble(in.readLine());
        I = Double.parseDouble(in.readLine());
        S0 = Double.parseDouble(in.readLine());
        S1 = Double.parseDouble(in.readLine());
        dstar = Double.parseDouble(in.readLine());
    } catch (Exception e) {
    System.out.println(e.toString());
    }
}

protected void readInputs(StringTokenizer st){
        dbar = Double.parseDouble(st.nextToken());
        nu = Double.parseDouble(st.nextToken());
        sigma = Double.parseDouble(st.nextToken());
        r = Double.parseDouble(st.nextToken());
        beta = Double.parseDouble(st.nextToken());
        alpha = Double.parseDouble(st.nextToken());
        d0 = Double.parseDouble(st.nextToken());
        I = Double.parseDouble(st.nextToken());
```

```
            S0 = Double.parseDouble(st.nextToken());
            S1 = Double.parseDouble(st.nextToken());
            dstar = Double.parseDouble(st.nextToken());
    }

    public String simulate(long seed, int N){
        String str = new String("");
        boolean exercise;
        int stepCount;
        double I;
        double v = d0, t, fv;
        double tstar = -1, count = 0,  totTstar = 0,  totVT = 0;
        Data ed = new Data(N);
        Data pd = new Data(N);
        Data C = new Data(N);

        if ((dstar < 0) || ((Double.isInfinite(dstar)))) {
                if (N < 2){
                str = str.concat(new String (df.format(tstar) + " "));
                }
                else {
                str = str.concat(new String (df.format(0) + " "));
                str = str.concat(new String (df.format(0)+ " "));
                str = str.concat(new String (df.format(0)+ " "));
                str = str.concat(new String (df.format(0)+ " "));
                str = str.concat(new String (df.format(-2)+ " "));
                str = str.concat(new String (df.format(0)+ " "));
                }
                return str;
```

```
        }

        try {
        if (seed > 0) sp.setSeed(seed);
        for (int n = 0; n < N; n++) {
            tstar = -1;
            sp.reset();
            exercise = false;
            stepCount = 0;
            I = 0;
            for(int i = 0; i <= 2*M; i++)  {
            sp.step();
            v = sp.value();
            t = sp.getTime();
            fv =  Math.min(v,S1)*Math.exp(-r*t);
            if ((i == 0) | (i == (2*M))) {
                I += fv;
            }
            else if ((i % 2) == 0) {
            I += 2*fv;
            }
            else {
            I += 4*fv;
            }

            // stopping times
            if (v >= dstar) {
                if (!exercise) {
                 tstar = t;
```

```
                exercise = true;
               }
            }
            }

            I *= dt/3.0;
            C.addPoint(I);

            if (tstar >= 0) {
            totTstar += tstar;
            ed.addPoint(tstar);
            pd.addPoint(1);
            count++;
            }
            else {
            pd.addPoint(0);
            }
            totVT += v;
        }

        if (N < 2){
            if (tstar < 0) {

            str = str.concat(new String (df.format(tstar) + " "));

            } else {
            str = str.concat(new String (df.format(tstar)+ " "));

            }
```

```
        }
        else {
        str = str.concat(new String (" " + df.format(dstar) + " "));
        str = str.concat(new String (df.format(pd.mean())+ " "));
        str = str.concat(new String (df.format(ed.mean())+ " "));
        str = str.concat(new String (
            df.format(1.96*Math.sqrt(ed.variance()/N)) + " & "));
        str = str.concat(new String (df.format(C.mean())+ " "));
        str = str.concat(new String (
            df.format(1.96*Math.sqrt(C.variance()/N))+ " \\ "));
        }
        } catch (Exception e) {
        System.out.println(e.toString());
        }
        return str;
    }

    // This function only applies for GBM case
    public double Cval(){
        CumulativeNormalDistribution cnd =
        new CumulativeNormalDistribution();
        double I = 0;
        for(int i = 0; i <= 2*M; i++) {
            double s = i*dt;
            double d1 = (Math.log(d0/S1)
                +(nu+Math.pow(sigma,2)/2)*s)/(sigma*Math.sqrt(s));
            double d2 = d1 - sigma*Math.sqrt(s);
            double fv =  d0*Math.exp((nu-r)*s)*cnd.value(-d1)
                + S1*Math.exp(-r*s)*cnd.value(d2);
```

```
            if ((i == 0) | (i == (2*M))) {
                I += fv;
            }
            else if ((i % 2) == 0) {
            I += 2*fv;
            }
            else {
            I += 4*fv;
            }
        }
        I *= dt/3.0;
        return I;
    }


    protected double dbar;
    protected double nu;
    protected double sigma;
    protected double r;
    protected double beta;
    protected double alpha;
    protected double d0;
    protected double I;
    protected double S0;
    protected double S1;
    protected double dstar;
    protected double t0 = 0;
    protected double dt;
    protected double T = 1000;
```

```
    protected int M = 50000;
    protected StochasticProcess sp;
    protected DecimalFormat df = new DecimalFormat("##.######");
}
```

Appendix D

MATLAB M-Files

The MATLAB language was used to implement the increasing capacity models in Chapters 5 and 6. It was also used extensively to plot all of the graphs in this thesis. The MATLAB M-files for the increasing capacity models are included in this appendix.

D.1 The Increasing Capacity Model in Chapter 5

The MATLAB language was used to implement the increasing capacity model in Chapter 5. The associated M-files are given below.

setglob.m

This function calculates some global variables.

```
function setglob(nuD,sigD,r,beta,alpha, D0, I, S0,S1)
global muD k k1 k2 A0 A1 B0 B1
muD = nuD/sigD - sigD/2;
k = (-muD + sqrt(muD^2 + 2*(r+alpha)))/sigD;
k1 =  (-muD + sqrt(muD^2+2*r))/sigD;
k2 =  (-muD -sqrt(muD^2+2*r))/sigD;
```

```
if nuD ~= r
A0 = (r-nuD*k2)/(r*(nuD-r)*(k2-k1)*S0^(k1-1));
B0 = (r-nuD*k1)/(r*(nuD-r)*(k2-k1)*S0^(k2-1));
A1 = (r-nuD*k2)/(r*(nuD-r)*(k2-k1)*S1^(k1-1));
B1 = (r-nuD*k1)/(r*(nuD-r)*(k2-k1)*S1^(k2-1));
else
A0 = ((log(S0)+1)*(1-k2)+1)/((0.5*sigD^2+r)*(k2-k1));
B0 = (0.5*sigD^2*S0^(k1-k2))/(r*(0.5*sigD^2+r)*(k1-k2));
A1 = ((log(S1)+1)*(1-k2)+1)/((0.5*sigD^2+r)*(k2-k1));
B1 = (0.5*sigD^2*S1^(k1-k2))/(r*(0.5*sigD^2+r)*(k1-k2))
end
```

phi.m

This function implements the $\phi(x)$ function.

```
function y = phi(S,nuD,sigD,r,x)
mu =  nuD/sigD - sigD/2;
k1 =  (-mu + sqrt(mu^2+2*r))/sigD
k2 =  (-mu -sqrt(mu^2+2*r))/sigD
if nuD ~= r
A = (r-nuD*k2)/(r*(nuD-r)*(k2-k1)*S^(k1-1));
B = (r-nuD*k1)/(r*(nuD-r)*(k2-k1)*S^(k2-1))
else
A = ((log(S)+1)*(1-k2)+1)/((0.5*sigD^2+r)*(k2-k1));
B = (0.5*sigD^2*S^(k1-k2))/(r*(0.5*sigD^2+r)*(k1-k2));
end


if x < S
    if nuD ~= r
```

```
        y = x/(nuD-r)  + A*x^k1;
    else
        y = (x*log(x))/(0.5*sigD^2+r)  + A*x^k1;
    end
else
    y = -S/r + B*x^k2;
end
```

dis.m

This function calculates the discount factor $E[e^{-\lambda\tau(x)}]$.

```
function y = dis(Y0,YS,mu,sigma,lambda)
k = (-mu + sqrt(mu^2 + 2*lambda))/sigma;
y = (Y0/YS)^k;
```

func.m

This function calculates the function $F(D, x)$.

```
function v = func(nuD,sigD,r,beta,alpha, D0, I, S0,S1,x)
global muD k k1 k2 A0 A1 B0 B1 C
y = phi(S0,nuD,sigD,r,x)* dis(D0,x,muD,sigD,r)
- phi(S1,nuD,sigD,r,x)* dis(D0,x,muD,sigD,r) ;
y =   y +   phi(S1,nuD,sigD,r,D0) - phi(S0,nuD,sigD,r,D0)
y = y + C
v = beta* y - I* dis(D0,x,muD,sigD,r+alpha);
```

deriv.m

This function calculates the function $F'(D, x)$.

```
function y= deriv(nuD,sigD,r,beta,alpha, D0, I, S0,S1,x)
global muD k k1 k2 A0 A1 B0 B1
iVal = k*I*(D0^k)*x^(-k-1);
if (x < S0)
bVal = 0;
else
if (x < S1)
if nuD ~= r
bVal = beta*(D0^k1)*(k1*(S0/r)*x^(-k1-1) + ((k1-1)/(nuD-r))*x^(-k1)
+ (k2-k1)*B0*x^(k2-k1-1));
else
bVal = beta*(D0^k1)*(k1*(S0/r)*x^(-k1-1) + (-2/(sigD^2+2*r))*x^(-k1)
+ (k2-k1)*B0*x^(k2-k1-1));
end
else
bVal = beta*(D0^k1)*(k1*((S0-S1)/r)*x^(-k1-1)
+ (k2-k1)*(B0-B1)*x^(k2-k1-1));
end
end
y = bVal + iVal;
```

fminus.m

This function calculates the function $f_{-}(x)$.

```
function y= fminus(nuD,sigD,r,beta,alpha, D0, I, S0,S1,x)
global muD k k1 k2 A0 A1 B0 B1
if nuD ~= r
y = k1*(S0/r)+ ((k1-1)/(nuD-r))*x + (k2-k1)*B0*x^(k2)
    +  k*(I/beta)*(D0/x)^(k-k1);
```

```
else
y = k1*(S0/r)- (2*x)/(sigD^2+2*r) + (k2-k1)*B0*x^(k2)
    +  k*(I/beta)*(D0/x)^(k-k1);
end
```

fplus.m

This function calculates the function $f_+(x)$.

```
function y= fplus(nuD,sigD,r,beta,alpha, D0, I, S0,S1,x)
global muD k k1 k2 A0 A1 B0 B1
y = k1*((S0-S1)/r) + (k2-k1)*(B0-B1)*x^(k2)
    + k*(I/beta)*((D0)^(k-k1))*(x^(-k+k1));
```

endval.m

This function finds an end value for the bisection method.

```
function y= endval(nuD,sigD,r,beta,alpha, D0, I, S0,S1)
global muD k k1 k2 A0 A1 B0 B1
x1 = ((k1*(S1-S0))/(3*r*(k2-k1)*(B0-B1)))^(1/k2);
x2 = D0*((beta*k1*(S1-S0))/(3*r*k*I))^(-1/(k-k1));
y = max(x1,x2);
```

trigger.m

This function finds the optimal trigger y^*.

```
function y= trigger(nuD,sigD,r,beta,alpha, D0, I, S0,S1)
d = fminus(nuD,sigD,r,beta,alpha, D0, I, S0,S1,S1)
if d < 0
    x1 = S0;
    x2 = S1;
```

```
    d1 = fplus(nuD,sigD,r,beta,alpha, D0, I, S0,S1,x2);
    d2 = d;
    while (abs(x1-x2) > 0.00001)
    x = 0.5*(x1+x2);
    d= fminus(nuD,sigD,r,beta,alpha, D0, I, S0,S1,x);
    if (d > 0)
        x1 = x;
        d1 = d;
    else
        x2 = x;
        d2 = d;
    end
    end
else
    x1 = S1;
    x2 = endval(nuD,sigD,r,beta,alpha, D0, I, S0,S1)
    d1 = d;
    d2 = fplus(nuD,sigD,r,beta,alpha, D0, I, S0,S1,x2);
    while (abs(x1-x2) > 0.00001)
    x = 0.5*(x1+x2);
    d= fplus(nuD,sigD,r,beta,alpha, D0, I, S0,S1,x);
    if (d > 0)
        x1 = x;
        d1 = d;
    else
        x2 = x;
        d2 = d;
    end
    end
```

```
end
y = x;
```

ihat.m

This function calculates the threshold $\widehat{I}$.

```
function y= ihat(nuD,sigD,r,beta,alpha, D0, I, S0,S1,x)
global muD k k1 k2 A0 A1 B0 B1
y = (beta/k)*(S1/D0)^(k-k1)*((k1*(S1-S0))/r + (k2-k1)*(B1-B0)*S1^k2);
```

bhat.m

This function calculates the threshold $\widehat{\beta}$.

```
function y= bhat(nuD,sigD,r,beta,alpha, D0, I, S0,S1)
global muD k k1 k2 A0 A1 B0 B1
y = (k*I)*(D0/S1)^(k-k1)*((k1*(S1-S0))/r
+ (k2-k1)*(B1-B0)*S1^k2)^(-1);
```

dhat.m

This function calculates the threshold $\widehat{D}$.

```
function y= dhat(nuD,sigD,r,beta,alpha, D0, I, S0,S1)
global muD k k1 k2 A0 A1 B0 B1
y = S1*((beta/(k*I))*((k1*(S1-S0))/r
+ (k2-k1)*(B1-B0)*S1^k2))^(1/(k-k1));
```

s1hat.m

This function calculates $f_+(S_1)$.

```
function y= s1hat(nuD,sigD,r,beta,alpha, D0, I, S0,S1)
global muD k k1 k2 A0 A1 B0 B1
if nuD ~= r
y = k1*((S0-S1)/r)
+ ((r-nuD*k1)/(r*(nuD-r)))*(S0^(-k2+1)-S1^(-k2+1))*S1^(k2)
+ k*(I/beta)*((D0)^(k-k1))*(S1^(-k+k1));
else
y = k1*((S0-S1)/r)
+ ((-sigD^2)/(r*(sigD^2+2*r)))*(S0^(-k2+1)-S1^(-k2+1))*S1^(k2)
+ k*(I/beta)*((D0)^(k-k1))*(S1^(-k+k1));
end
```

s0trig.m

This function calculates the threshold $\widehat{S}_0$.

```
function y= s0trig(nuD,sigD,r,beta,alpha, D0, I, S0,S1)
    x1 = 0;
    x2 = S1;
    d1 = s1hat(nuD,sigD,r,beta,alpha, D0, I, x1,S1);
    d2 = s1hat(nuD,sigD,r,beta,alpha, D0, I, x2,S1);
    while (abs(x1-x2) > 0.00001)
    x = 0.5*(x1+x2);
    d= s1hat(nuD,sigD,r,beta,alpha, D0, I, x,S1);
    if (d < 0)
        x1 = x;
        d1 = d;
    else
        x2 = x;
        d2 = d;
```

```
    end
    end
y = x;
```

s1trig.m

This function calculates the threshold $\widehat{S}_1$.

```
function y= s1trig(nuD,sigD,r,beta,alpha, D0, I, S0,S1)
    x1 = S0;
    x2 = 20;
    d1 = s1hat(nuD,sigD,r,beta,alpha, D0, I, S0,x1);
    d2 = s1hat(nuD,sigD,r,beta,alpha, D0, I, S0,x2);
    while (abs(x1-x2) > 0.000001)
    x = 0.5*(x1+x2);
    d= s1hat(nuD,sigD,r,beta,alpha, D0, I, S0,x);
    if (d > 0)
        x1 = x;
        d1 = d;
    else
        x2 = x;
        d2 = d;
    end
    end
y = x;
```

D.2 The Increasing Capacity Model in Chapter 6

The MATLAB language was used to implement the increasing capacity model in Chapter 6. The associated M-files are given below.

kumm.m

This function implements the $M(a, b, x)$ function.

```
function y = kumm(a1,b1,x1)
a = a1;
b = b1;
x = x1;
n = 1;
mul = 1;
val = 0;
for i = 1:2000
val = val + mul;
mul = mul*a*x/(b*n);
a = a+1;
b = b+1;
n = n+1;
end
y = val;
```

umm.m

This function implements the $U(a, b, x)$ function.

```
function y = umm(a,b,x)
y = pi/sin(pi*b);
k1 = kumm(a,b,x);
k2 = kumm(1+a-b,2-b,x);
k1 = k1/gamma(1+a-b);
k1 = k1/gamma(b);
k2 = kumm(1+a-b,2-b,x);
```

```
k2 = k2 /gamma(a);
k2 = k2 / gamma(2-b);
y = y*(k1 -x^(1-b)*k2);
```

yumm.m

This function implements the $Y(a, b, x)$ function.

```
function y = yumm(a1,b1,x1)
a = a1;
b = b1;
x = x1;
n = 1;
mul = 1;
val = 0;
for i = 1:2000
val = val + mul/(n-1+a1);
mul = mul*a*x/(b*n);
a = a+1;
b = b+1;
n = n+1;
end
y = val;
```

psi1.m

This function implements the $\psi_1(x)$ function.

```
function y = psi1(x)
global a b a1 b1 muD k k1 k2 A0 A1 B0 B1
y = x^a*kumm(a,b,k*x);
```

dpsi1.m

This function implements the $\psi_1'(x)$ function.

```
function y = dpsi1(x)
global a b a1 b1 muD k k1 k2 A0 A1 B0 B1
y = a*x^(a-1)*kumm(a+1,b,k*x);
```

psi3.m

This function implements the $\psi_3(x)$ function.

```
function y = psi3(x)
global a b a1 b1 muD k k1 k2 A0 A1 B0 B1
y = x^a1*kumm(a1,b1,k*x);
```

dpsi3.m

This function implements the $\psi_3'(x)$ function.

```
function y = dpsi3(x)
global a b a1 b1 muD k k1 k2 A0 A1 B0 B1
y = a1*x^(a1-1)*kumm(a1+1,b1,k*x);
```

psi4.m

This function implements the $\psi_4(x)$ function.

```
function y = psi4(x)
global a b a1 b1 muD k k1 k2 A0 A1 B0 B1
y = x^a*umm(a,b,k*x);
```

dpsi4.m

This function implements the $\psi_4'(x)$ function.

```
function y = dpsi4(x)
global a b a1 b1 muD k k1 k2 A0 A1 B0 B1
y = a*(a+1-b)*x^(a-1)*umm(a+1,b,k*x);
```

psi8.m

This function implements the $\psi_8(x)$ function.

```
function y = psi8(x)
global a b a1 b1 muD k k1 k2 A0 A1 B0 B1
y = x^(b-a)*yumm(b-a,b,-k*x);
```

capphi.m

This function implements the $\Phi(x)$ function.

```
function y = capphi(x)
global a b a1 b1 muD k k1 k2 A0 A1 B0 B1
y = x*kumm(1+a-b,2-b,k*x)*yumm(b-a,b,-k*x)
- x*kumm(a,b,k*x)*yumm(1-a,2-b,-k*x);
```

dcapphi.m

This function implements the $\Phi'(x)$ function.

```
function y = dcapphi(x)
global a b a1 b1 muD k k1 k2 A0 A1 B0 B1
y =  (1+a-b)*kumm(2+a-b,2-b,k*x)*yumm(b-a,b,-k*x)
- a*kumm(a+1,b,k*x)*yumm(1-a,2-b,-k*x);
```

setglob.m

This function calculates some global variables.

```
function setglob(dbar,nuD,sigD, r, beta,alpha, D0, I, S0, S1)
global muD a0 a a1 b b1 k C C1 B0 B1 A0 A1 trig n
muD = (nuD)/sigD - sigD/2;
a0 = (-muD -sqrt(muD^2+2*r))/sigD;
a =  (-muD + sqrt(muD^2+2*r))/sigD;
a1 = (-muD + sqrt(muD^2 + 2*(r+alpha)))/sigD;
b = 2*a + 2*(nuD)/(sigD^2);
b1 = 2*a1 + 2*(nuD)/(sigD^2);
C1 = 1/r;
if dbar == -1
k = 0;
C = 1/(nuD-r);
A0 = (r-nuD*a0)/(r*(nuD-r)*(a0-a)*S0^(a-1));
B0 = (r-nuD*a)/(r*(nuD-r)*(a0-a)*S0^(a0-1));
A1 = (r-nuD*a0)/(r*(nuD-r)*(a0-a)*S1^(a-1));
B1 = (r-nuD*a)/(r*(nuD-r)*(a0-a)*S1^(a0-1));
else
C = 2/(sigD^2);
eta = nuD/dbar;
k = (2*eta)/(sigD^2);
S = S0 ;
A = -(gamma(a)/gamma(b))*k^(b-1)*C*(gamma(1-b)/gamma(1+a-b))
*S^(b-a)*yumm(b-a,b,-k*S);
A = A + (gamma(a)/gamma(b))*k^(b-1)*(S^(1+b-2*a)*dpsi4(S))/(r*exp(k*S));
A = A + (C/(1-b))*S^(1-a)*yumm(1-a,2-b,-k*S);
A0 = A;
```

```
S= S1;
A = -(gamma(a)/gamma(b))*k^(b-1)*C*(gamma(1-b)/gamma(1+a-b))
*S^(b-a)*yumm(b-a,b,-k*S);
A = A + (gamma(a)/gamma(b))*k^(b-1)*(S^(1+b-2*a)*dpsi4(S))/(r*exp(k*S));
A = A + (C/(1-b))*S^(1-a)*yumm(1-a,2-b,-k*S);
A1 = A;
B0  = -C*S0^(b-a)*yumm(b-a,b,-k*S0)
+ S0^(1+b-2*a)*dpsi1(S0)*exp(-k*S0)/r;
B0  = (gamma(a)/gamma(b))*k^(b-1)* B0;
B1  = -C*S1^(b-a)*yumm(b-a,b,-k*S1)
+ S1^(1+b-2*a)*dpsi1(S1)*exp(-k*S1)/r;
B1  = (gamma(a)/gamma(b))*k^(b-1)* B1;
end
```

phi0.m

This function implements the $\phi_0(x)$ function.

```
function y = phi0(x)
global muD k a0 a b a1 b1 A0 A1 B0 B1 C C0 C1 S0 S1
if k == 0
    if x < S0
        y = C*x  + A0*x^a;
    else
        y = -C1*S0 + B0*x^a0;
    end
else
    if x < S0
        y = C*capphi(x)/(1-b) + A0*psi1(x);
    else
```

```
        y = -C1*S0 + B0*psi4(x);
    end
end
```

phi1.m

This function implements the $\phi_1(x)$ function.

```
function y = phi1(x)
global muD k a0 a b a1 b1 A0 A1 B0 B1 C C0 C1 S0 S1
if k == 0
    if x < S1
        y = C*x  + A1*x^a;
    else
        y = -C1*S1 + B1*x^a0;
    end
else
    if x < S1
        y = C*capphi(x)/(1-b) + A1*psi1(x);
    else
        y = -C1*S1 + B1*psi4(x);
    end
end
```

dis.m

This function calculates the discount factor $E[e^{-r\tau(x)}]$.

```
function y = dis(Y0,YS)
global a b k
if k == 0
```

```
    y = (Y0/YS)^a;
else
    y = (Y0/YS)^a*kumm(a,b,k*Y0)/kumm(a,b,k*YS);
end
```

dis1.m

This function calculates the discount factor $E[e^{-(r+\alpha)\tau(x)}]$.

```
function y = dis(Y0,YS)
global a1 b1 k
if k == 0
    y = (Y0/YS)^a1;
else
    y = (Y0/YS)^a1*kumm(a1,b1,k*Y0)/kumm(a1,b1,k*YS);
end
```

func.m

This function implements the $G(D,x)$ function.

```
function v = func(beta, I, D0,x)
global a b a1 b1 muD k k1 k2 A0 A1 B0 B1 C C0 C1 S0 S1
y = phi0(x)*dis(D0,x) - phi1(x)*dis(D0,x) ;
y =   y +   phi1(D0) - phi0(D0);
y = y + C1*S1;
v = beta* y - I* dis1(D0,x);
```

deriv.m

This function implements the $F'(D,x)$ function.

```
function y= deriv(beta, I, D0, x)
global k a0 a b a1 C C1 B0 B1 S0 S1
if (k==0)
iVal = a1*I*(D0^a1)*x^(-a1-1);
else
iVal = I*psi3(D0)*dpsi3(x)/(psi3(x)^2);
end
if (x < S0)
bVal = 0;
else
if (x < S1)
if (k==0)
bVal = beta*(D0^a)*(a*C1*S0*x^(-a-1) + C*(a-1)*x^(-a)
+ (a0-a)*B0*x^(a0-a-1));
else
bVal = beta*psi1(D0)*(C1*S0*dpsi1(x)/(psi1(x)^2)
- C*x^(2*a-b)*exp(k*x)*psi8(x)/(psi1(x)^2)
-  gamma(b)*k^(1-b)*B0*x^(2*a-b)*exp(k*x)/(gamma(a)*psi1(x)^2));
end
else
if (k==0)
bVal = beta*(D0^a)*(a*C1*(S0-S1)*x^(-a-1) + (a0-a)*(B0-B1)*x^(a0-a-1));
else
bVal = beta*psi1(D0)*(C1*(S0-S1)*dpsi1(x)/(psi1(x)^2)
- gamma(b)*k^(1-b)*(B0-B1)*x^(2*a-b)*exp(k*x)/(gamma(a)*psi1(x)^2));
end
end
end
y = bVal + iVal;
```

trigger.m

This function finds the optimal trigger y^*.

```
function y= trigger(beta, I, D0)
global S0 S1
d = deriv(beta, I, D0, S1)
if d < 0
    x1 = S0;
    x2 = S1;
    d1 = deriv(beta, I, D0, x1);
    d2 = d;
else
    x1 = S1;
    x2 = 100;
    d1 = d;
    d2 = deriv(beta, I, D0, x2);
end
    while (abs(x1-x2) > 0.00001)
    x = 0.5*(x1+x2);
    d= deriv(beta, I, D0, x);
    if (d > 0)
        x1 = x;
        d1 = d;
    else
        x2 = x;
        d2 = d;
    end
    end
y = x;
```

Bibliography

[1] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions.* Dover Publications, New York, 1970.

[2] James Alleman. A new view of telecommunications economics. *Telecommunications Policy: The International Journal on Knowledge Infrastructure Development Management and Regulation*, 26(January 2002):87–92, 2002.

[3] Louis Bachelier. Théorie de la spéculation. *Les Annales Scientifiques de L'École Normale Supérieure*, 7:21–86, 1900.

[4] C. Baldwin. Competing for capital in a global environment. *Midland Corporate Finance Journal*, 1:43–64, 1987.

[5] Sanjai Bhagat. Real options applications in the telecommunications industry. In James Alleman and Eli Noam, editors, *The New Investment Theory of Real Options and its Implications for Telecommunications Economics.* Kluwer Academic Publishers, Boston, 1999.

[6] P. Bjerksund and S. Ekern. Contingent claim evaluation of mean-reverting cash flows in shipping. In Lenos Trigeorgis, editor, *Real Options in Capital Investment: models, strategies, and applications.* Praeger, Westport, Conn., 1995.

[7] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81:637–659, 1973.

[8] Phelim P. Boyle. Options: A monte carlo approach. *Journal of Financial Economics*, 4:323–338, 1977.

[9] Michael J. Brennan and Eduardo S. Schwartz. The valuation of American put options. *The Journal of Finance*, 32:449–462, 1977.

[10] Mark Broadie and Paul Glasserman. Pricing American-style securities using simulation. *Journal of Economic Dynamics and Control*, 21:1323–1352, 1997.

[11] Paul Budde. 2005-2006 Australia telecoms industry - overview and statistics report. Technical report, Paul Budde Communication Pty Ltd., 15 August 2005.

[12] Marc Chesney, Henri Loubergé, and Stéphane Villeneuve. Long term risk management of nuclear waste: a contingent claim analysis. *Journal of Economic Dynamics and Control*, 27(1):157–180, 2002.

[13] Ephraim Clark and Joshy Z. Easaw. *Optimal Network Access Pricing for Natural Monopolies when Costs are Sunk and Revenues are Uncertain.* Working paper, University of Bath, 2003.

[14] Richard N. Clarke. Rethinking the implications of "real options" theory for the U.S. local telephone industry. In James Alleman and Eli Noam, editors, *The New Investment Theory of Real Options and its Implications for Telecommunications Economics.* Kluwer Academic Publishers, Boston, 1999.

[15] K. G. Coffman and A. M. Odlyzko. Internet growth: Is there a Moore's law for data traffic? In P. M. Pardalos J. Abello and M. G. C. Resende, editors, *Handbook of Massive Data Sets.* Kluwer, Dordrecht, 2002.

[16] Samuel D. Conte and Carl de Boor. *Elementary Numerical Analysis: An Algorithmic Approach.* McGraw-Hill, New York, third edition, 1980.

[17] John C. Cox, Stephen A. Ross, and Mark Rubinstein. Option pricing: A simplified approach. *Journal of Financial Economics*, 7:229–263, 1979.

[18] Rose-Anne Dana and Monique Jeanblanc. *Financial Markets in Continous Time.* Springer, New York, 2003.

[19] Yann d'Halluin, Peter A. Forsyth, and Kenneth R. Vetzal. *Wireless Network Capacity Investment.* Working paper, University of Waterloo, 2004.

[20] Yann d'Halluin, Peter A. Forsyth, and Kenneth R. Vetzal. Managing capacity for networking. *IEEE/ACM Transactions on Networking*, 10(4):579–588, August, 2002.

[21] A. K. Dixit. *The Art of Smooth Pasting.* Routledge, London, 2003.

[22] Avinash K. Dixit and Robert S. Pindyck. *Investment Under Uncertainty.* Princeton University Press, Princeton, N.J., 1994.

[23] Nicholas Economides. Real options and the cost of local telecommunications network. In James Alleman and Eli Noam, editors, *The New Investment Theory of Real Options and its Implications for Telecommunications Economics.* Kluwer Academic Publishers, Boston, 1999.

[24] Robert J. Elliott and John van der Hoek. Pricing non tradable assets: Duality methods. In Rene Carmona, editor, *Indifference Pricing.* Princeton University Press, Princeton, N.J., to appear 2007.

[25] Robert J. Elliott and P. Ekkehard Kopp. *Mathematics of Financial Markets, second edition.* Springer, New York, 2005.

[26] G. Gilder. Fiber keeps its promise: Get ready, bandwidth will triple each year for the next 25. In *Forbes*, April, 1997.

[27] G. Gilder. The bandwidth tidal wave. In *Forbes ASAP*, December, 1994.

[28] Steven R. Grenadier. Option exercise games: An application to the equilibrium investment strategies of firms. *Review of Financial Studies*, 15(3):691–721, 2002.

[29] J. Michael Harrison. *Brownian Motion and Stochastic Flow Systems.* John Wiley & Sons, New York, 1985.

[30] Jerry Hausman. The effect of sunk costs in telecommunication regulation. In James Alleman and Eli Noam, editors, *The New Investment Theory of Real Options and its Implications for Telecommunications Economics.* Kluwer Academic Publishers, Boston, 1999.

[31] Jerry Hausman. Rejoinder. In James Alleman and Eli Noam, editors, *The New Investment Theory of Real Options and its Implications for Telecommunications Economics.* Kluwer Academic Publishers, Boston, 1999.

[32] Vicky Henderson. Valuing real options without a perfect spanning asset. In *8th Annual Real Options Conference*, Montréal Canada, 2004.

[33] John C. Hull. *Options, Futures & Other Derivatives, Fourth edition.* Prentice Hall, Upper Saddle River, N.J., fifth edition, 2003.

[34] Ioannis Karatzas and Steven E. Shreve. *Brownian Motion and Stochastic Calculus.* Springer-Verlag, New York, 1991.

[35] Ioannis Karatzas and Steven E. Shreve. *Methods of Mathematical Finance.* Springer-Verlag, New York, 1998.

[36] Chris Kenyon and Giorgos Cheliotis. Dark fibre valuation. *The Engineering Economist*, 47(3):264–307, 2002.

[37] Peter E. Kloeden, Eckhard Platen, and Henri Schurz. *Numerical Solution of SDE Through Computer Experiments.* Springer-Verlag, Berlin; New York, 1994.

[38] S. G. Kou and Hui Wang. Option pricing under a double exponential jump diffusion model. *Management Science*, 50(9):1178–1192, 2004.

[39] N. Kulatilaka. Valuing the flexibility of flexible manufacturing systems. *IEEE Transactions in Engineering Management*, 35(4):250–257, 1988.

[40] Jean-Jacques Laffont and Jean Tirole. *Competition in Telecommunications.* The MIT Press, Cambridge, Mass.; London, 2000.

[41] J. Lassila. *Real Options in Telecommunications Capacity Markets.* Masters thesis, Helsinki University of Technology, 2001.

[42] Averill M. Law and W. David Kelton. *Simulation Modeling and Analysis.* McGraw-Hill, New York, second edition, 1991.

[43] Robert McDonald and Daniel Siegel. The value of waiting to invest. *Quarterly Journal of Economics*, 101:707–728, 1986.

[44] Robert C. Merton. Theory of rational option pricing. *The Bell Journal of Economics and Management Science*, 4:141–183, 1973.

[45] Gordon E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8):114–117, April, 1965.

[46] P. A. Morris, E. O. Teisburg, and A. L. Kolbe. When choosing R&D projects, go with long shots. *Research-Technology Management*, 34(1):35–40, 1991.

[47] M. Musiela and T. Zariphopoulou. *Pricing and Risk Management of Derivatives Written on Non-Traded Assets.* Working paper, BNP Paribas, London, 2001.

[48] A. M. Odlyzko. Comments on the Larry Roberts and Caspian networks study of internet traffic growth. In *The Cook Report on the Internet*, pages 12–15, December, 2001.

[49] Michael D. Pelcovits. Application of real options theory to TELRIC models: Real trouble or red herring. In James Alleman and Eli Noam, editors, *The New Investment Theory of Real Options and its Implications for Telecommunications Economics.* Kluwer Academic Publishers, Boston, 1999.

[50] Philip Protter. *Stochastic Integration and Differential Equations.* Springer-Verlag, Berlin; New York, 1992.

[51] Albert L. Rabenstein. *Introduction to Ordinary Differential Equations.* Academic Press, New York, 1972.

[52] Richard J Rendleman and Brit J. Bartter. Two-state option pricing. *The Journal of Finance*, 34(5):1092–1110, 1979.

[53] Daniel Revuz and Marc Yor. *Continuous Martingales and Brownian Motion.* Springer, Berlin; New York, third edition, 1999.

[54] Sheldon M. Ross. *Simulation.* Academic Press, San Diego, second edition, 1997.

[55] Paul A. Samuelson. Rational theory of warrant pricing. *Industrial Management Review*, 6(2):13–40, 1965.

[56] Steven E. Shreve. *Stochastic Calculus for Finance II: Continuous Time Models.* Springer, New York, 2004.

[57] Gordon Sick. *Capital Budgeting with Real Options.* Monograph Series in Finance and Economics, Monograph 1989-3. Salomon Brothers Center for the Study of Financial Institutions, Leonard N. Stern School of Business, New York University, 1989.

[58] John P. Small. *Real Options and the Pricing of Network Access.* Working paper, University of Auckland, 1998.

[59] James E. Smith and Kevin F. McCardle. Valuing oil properties: Integrating option pricing and decision analysis approaches. *Operations Research*, 46(2):198–217, 1998.

[60] Lenos Trigeorgis. *Real Options: Managerial Flexibility and Resource Allocation.* The MIT Press, Cambridge, Mass., 1996.

[61] Lenos Trigeorgis. Real options: An overview. In Eduardo S. Schwartz and Lenos Trigeorgis, editors, *Real Options and Investment Under Uncertainty.* The MIT Press, Cambridge, Mass., 2001.

[62] E. W. Weisstein. Mathworld. *http://mathworld.wolfram.com.*

[63] Wikipedia. The free encylopedia. *http://en.wikipedia.org.*

[64] Paul Wilmott. *Paul Wilmott on Quantitative Finance.* Wiley, Chichester, 2000.