Pricing Convertible Bonds using Partial Differential Equations

by

Lucy Xingwen Li

A thesis submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

Abstract

Pricing Convertible Bonds using Partial Differential Equations

Lucy Xingwen Li

Master of Science

Graduate Department of Computer Science

University of Toronto

2005

A Convertible Bond (CB) is a corporate debt security that gives the holder the right to exchange future coupon payments and principal repayment for a prescribed number of shares of equity. Thus, it has both an equity part and a fixed-income part, and may contain some additional features, such as callability and puttability.

In this paper, we study the model for valuing Convertible Bonds with credit risk originally developed by Kostas Tsiveriotis and Chris Fernandes (TF). The Convertible Bond is a derivative of the stock price, and the pricing model developed by TF is based on a free boundary value problem associated with a pair of parabolic Partial Differential Equations (PDEs) with discontinuities at the time points when there is a coupon payment, or when the bond is converted, or when it is called back (purchased) by the issuer, or when it is put (sold) to the issuer. We explore the possible derivation of the TF model and study the convergence of several numerical methods for solving the free boundary value problem associated with the TF model. In particular, we consider the Successive Over-Relaxation (SOR) iteration and a penalty method for solving the linear complementarity problem used to handle the free boundary. Special emphasis is given to the effectiveness of the numerical scheme as well to the treatment of discontinuities.

Acknowledgements

I would like to express my deep appreciation and sincere gratitude to my supervisors Professor Ken Jackson and Professor Christina Christara for their willingness to take me into the Financial Computation Group, and also for their passionate support, expert guidance, and continuous encouragement during my study and research. I also would like to thank Dr. Izzy Nelken for suggesting this topic, his valuable advice and his constructive comments. I would like to thank Professor Robert Almgren for his valuable guidance in our summer reading group, and his generous assistance with the model derivation.

I was also lucky to be able to associate myself with the talented and hard working members of the Numerical Analysis group. I would like to express my special thanks to Xiaofang Ma for his assistance and encouragement.

Finally I want to thank my parents and my daughter for their life-long love and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A bond is a financial instrument or contract which is paid for up-front and yields a known amount on a known date in the future. A bond may also pay a known cash dividend at fixed times during the life of the contract. The cash dividend is often called a coupon, and is usually paid semiannually or annually. Depending on the issuer, bonds are categorized as corporate bonds or government bonds. The main reason a bond is issued is to raise capital.

A Convertible Bond is a corporate bond with the additional feature that the bond owner can exchange (*convert*) the bond into a specified asset, e.g. the company's stocks, at some time in the future. That is, the owner either receives periodic coupon payments and a prescribed amount at a prescribed time when the bond expires, or the owner converts the bond some time before it expires and forgoes the coupon payments and principal repayment. A Convertible Bond may be associated with a *call* option, in the sense that the company that issued the bond may have the right to buy (call) it back. Likewise, a Convertible Bond may be associated with a *put* option, in the sense that the holder may return (sell) the bond back to the issuer for a specified amount. Usually, the holder of a Convertible Bond may convert it even after it is called. Therefore, a Convertible Bond may involve many features, and its valuation (*i.e., determining a fair* *price for the bond*) is complex.

Today, Convertible Bonds are very common, composing about 10% of all debt in the USA (this is the average ratio of convertible to total debt between 1900 and 1993 [8]). It is an indirect means to add equity to the capital structure, allowing a firm to borrow more cheaply than if it issued a non-Convertible Bond [13]. However, a Convertible Bond is very difficult to model and evaluate because its pricing involves both equity and debt, as well as possible embedded call and put option features. Thus, pricing a Convertible Bond involves many factors, such as the equity price, the maturity, the interest rate, the volatility, the conversion ratio, and the exercise price. Currently, many researchers are studying Convertible Bond pricing.

There are essentially two approaches to modelling a Convertible Bond [8]: one is based on the firm value, the other is based on the equity value. The firm value approach uses the modern Black-Scholes-Merton model. It assumes that the value of the firm as a whole is composed of equity and Convertible Bonds, and it models the value of the firm as a geometric Brownian motion. For example, Ingersoll decomposes the value of a non-callable Convertible Bond (CB) into a straight bond K (with the same principal as the Convertible Bond) and a warrant with an exercise price equal to the face value of the bond, i.e. $CB = K + \max(\gamma V_T - K, 0)$ where V_T is the value of the company at T and γ is the fraction of the equity that the bond holders obtain if they convert. Ingersoll then generalizes his result to price Convertible Bonds with calls. Firm value Convertible Bond models have also been developed by Brennan and Schwartz [3], Nyborg [13], and as mentioned in [8] by Merton and Geske. On the other hand, the equity-value approach to pricing Convertible Bonds is based on the equity, rather than the value of the firm. The equity is modelled by geometric Brownian motion and uses the Black-Scholes equation. The equity value Convertible Bond models include those developed by Tsiveriotis and Fernandes [16], Ho and Pfeffer [9], and as mentioned in [8] by Goldman Sachs, Davis and Lischka, and Quinlan. The firm value is not directly observable and has to be inferred;

moreover, the true complex nature of the capital structure of the firm makes it difficult to model. On the other hand, the price of the equity is explicitly observable in the market. However, when a firm encounters financial difficulty, the firm value approach becomes easier. Under the risk-neutral measure, the equity price follows a stochastic process. The Convertible Bond value is determined by the maximum of the following values: the put price, the conversion value, and the minimum of the call price and the current Convertible Bond value. Cash-flows are valued differently depending on whether they are related to equity or debt.

The model developed by Kostas Tsiveriotis and Chris Fernandes (henceforth referred to as TF)[16] is based on the equity value approach. The model consists of two coupled Partial Differential Equations (PDEs). The PDEs are linear, thus all the difficulty is in the treatment of the boundary conditions and the treatment of the discontinuities associated with the PDEs. This research paper explores accurate and efficient numerical methods for solving these PDEs with particular attention to the method for handling the boundary conditions and the discontinuities. We investigate three Finite Different Methods (FDMs): the explicit method, the fully implicit method, and the Crank-Nicolson method. The explicit method applies the constraints directly; it is a first-order method, and it is conditionally stable; therefore the convergence speed is slow. The fully implicit method is also first-order method, but it is unconditionally stable. The Crank-Nicolson method can be viewed as a combination of the two methods; it is a second-order implicit method and unconditionally stable. For the implicit schemes, two iterative approaches to handling the free boundary conditions are considered. One approach is to view the problem as a linear complementarity problem, and then use a Projected Successive Over-Relaxation (PSOR) technique to solve the discrete algebraic equations. A second approach is to view the problem as a nonlinear algebraic system, where the nonlinear constraint is approximated by a penalty method. The resulting system of nonlinear algebraic equations is then solved using a Newton iteration.

In order to gain some insight into the two approaches outlined above, we start by considering the simpler problem of pricing an American option, which has a free boundary similar to that associated with a Convertible Bond. This gives some insight into the more complex problem of pricing a Convertible Bond.

A Convertible Bond may be converted some time during its life. We consider the American-style Convertible Bond which can be converted any time during its life. Other types of Convertible Bonds have restrictions on when they can be converted. They can be viewed as special cases of the American-style Convertible Bond, and are therefore easier to evaluate.

The organization of this thesis is as follows. Chapter 2 introduces the TF model and its derivation. Chapter 3 presents finite difference discretizations of the PDEs associated with the TF model. Chapter 4 explores the two approaches for handling the free boundary associated with the TF model: the Projected Successive Over-Relaxation (PSOR) method and the penalty method. Chapter 5 presents the numerical results, and discusses the advantages and disadvantages of the methods considered. Chapter 6 concludes and discusses some possible future work.

Chapter 2

Mathematical Model

This chapter introduces some models for pricing Convertible Bonds. To gain some insight into this complex problem, we first consider the similar, but simpler, problem of pricing an American option. We then discuss the TF model [16] for pricing a Convertible Bond, as well as the related model in the paper of Ayache, Forsyth, and Vetzal [1]. Finally we provide possible derivations for the TF model, since we could not find these in the literature.

2.1 The American Option

The American option is a contract that gives the holder the right, but not the obligation, to buy or sell the underlying asset (e.g., a stock) at a specified price in the future. The option can be exercised any time before or at the maturity of the option, whereas a European option can be exercised at maturity only. The added flexibility of an American option typically makes it more valuable than the corresponding European option. However, this added flexibility also makes an American option considerably more difficult to price than the corresponding European option. For example, some European options can be evaluated analytically, while American options typically have to be priced numerically. We take an American put option as an example.

Let S denote the price of the underlying asset, typically the stock price, and let E denote the exercise price. Let T be the maturity time of the option, and let t denote the current time: $0 \leq t \leq T$. The option price V is a function of S and t. The standard approach for pricing an American option is associated with a free boundary problem [18], and the exercise boundary $S_f(t)$ generally varies with time. The American put option should be exercised if $S < S_f(t)$ and held otherwise. At any time during the life of the option, the value of a put option satisfies

$$V(S,t) \geq \max(E - S, 0). \tag{2.1}$$

In addition, the option value satisfies the following differential inequality associated with the Black-Scholes Partial Differential Equation [18]:

$$\frac{\partial V}{\partial t} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV \leq 0. \tag{2.2}$$

At any given time, one of (2.1) or (2.2) must be an equality. The final condition is

$$V(S,T) = \max(E - S, 0), \tag{2.3}$$

and the boundary conditions are

$$\begin{aligned} V(0,t) &= E, \\ V(S,t) &\sim 0 \qquad \text{as} \quad S \to \infty. \end{aligned} \tag{2.4}$$

We can reformulate the problem above as a linear complementarity problem. To this end, we introduce the notation

$$\mathcal{L}V = \frac{\partial V}{\partial t} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV, \tag{2.5}$$

and let

$$V^*(S,t) = \max(E - S, 0), \tag{2.6}$$

denote the payoff function. The option can either continue to be held ($\mathcal{L}V = 0$) or be exercised ($V = V^*$); therefore, at least one equation holds. Thus, the American option price $V(S,t)$ satisfies the linear complementarity problem

$$\begin{pmatrix} \mathcal{L}V = 0 \\ (V - V^*) \geq 0 \end{pmatrix} \vee \begin{pmatrix} \mathcal{L}V \leq 0 \\ (V - V^*) = 0 \end{pmatrix}$$

together with the final condition (2.3) and the boundary conditions (2.4). In (2.1), the symbol $\vee$ means that either the two conditions in the left bracket hold or the two conditions in the right bracket hold.

2.2 The TF Model

A Convertible Bond (CB) has both an equity part and a fixed-income part. In general, the CB is regarded as a derivative of the underlying equity and interest rate, and can be accurately valued only by simultaneously pricing the equity and fixed-income parts. Considering the exposure to different credit risks, or default risks, the TF model views the value of a CB as two components. It introduces a new hypothetical security, the "Cash-Only" part of the CB, abbreviated as COCB. The holder of the COCB is entitled to all cashflows, and no equity flows, that an optimally behaving holder of the corresponding CB would receive. This part is related to the future cash payments: the coupon, principal repayment, and, when coupled with put provisions, the cash payment to the holder if the holder sells the CB back to the issuer. Clearly, these cashflows depend on the issuer's timely access to the required cash amounts, and thereby introduce credit risk. However, the equity part has zero default risk, given that the issuer can always deliver its own stock.

To evaluate the credit risk, the TF model introduces an effective credit spread, which can be simply approximated. CBs should be viewed and valued as derivatives of the underlying equity and interest rate, and, because the value of the future cash payments that a rational CB holder will choose to receive is itself a derivative of the underlying equity and interest rates, the value of the COCB is also a derivative security of the same underlyings.

Let U represent the CB value, and B represent the COCB value. Then $U - B$ represents the value of the CB related to payments in equity, and it should be discounted using the risk-free interest rate.

The TF model is described using two coupled parabolic partial differential equations,

$$\frac{\partial B}{\partial t} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 B}{\partial S^2} + r_g S\frac{\partial B}{\partial S} - (r + r_c)B + f(t) = 0, \tag{2.7}$$

$$\frac{\partial U}{\partial t} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 U}{\partial S^2} + r_g S\frac{\partial U}{\partial S} - r(U - B) - (r + r_c)B + f(t) = 0, \tag{2.8}$$

where S is the price of the underlying stock, r the risk-free interest rate, r_g the growth rate of the stock, r_c the observable credit spread between the convertible and the non-convertible bonds of the same issuer for maturities similar to that of the CB, and $f(t)$ describes various predetermined external flows — in cash or equity – to the derivative. For example, for a bond paying a coupon of c_j at time t_j, we have

$$f(t) = \sum c_j \delta(t - t_j), \tag{2.9}$$

where δ is the Dirac function. We will ignore $f(t)$ for now, and leave the handling of coupon payments to the later chapters (Chapter 3 and 4). Equations (2.7) and (2.8) differ only in the discounting terms $(r + r_c)B$ and $r(U - B) + (r + r_c)B$, which reveal the different credit treatment of cash payments and the equity upside. The derivation of equations (2.7) and (2.8) is described in the next section.

Now we discuss in detail the final, boundary and other conditions that the solutions must satisfy due to conversion, callability, and puttability.

Assume that the bond has the face value of F, and a coupon payment of $c_j = C$ semiannually. Let T be the maturity time of the Convertible Bond, and assume that the bond can be converted any time during its life. The conversion ratio is κ, which means the bond can be converted into κ shares of the company's stock. It is callable at the price B_c, and it is puttable at the price B_p. Note both B_c and B_p are functions of t due to the accrued interest which will be addressed in Chapter 3.

At maturity, the bond holder may choose either to be repaid the principal F or given κ shares of the company's stock. So the final condition is

$$B(S,T) = \begin{cases} F+C & \text{if } F+C \geq \kappa S \\ 0 & \text{if } F+C < \kappa S. \end{cases}$$

$$U(S,T) = \begin{cases} F+C & \text{if } F+C \geq \kappa S \\ \kappa S & \text{if } F+C < \kappa S, \end{cases}$$

Without loss of generality, when the CB is not puttable, we can set $B_p = 0$, and, when it is not callable, we can set $B_c = \infty$. The bond price should satisfy the following constraint

$$U = \max(B_p, \kappa S, \min(B_c, \hat{U})), \tag{2.10}$$

where $\hat{U}$ is the "continuous value" of the CB, which means that none of the events of convertibility, puttability, and callability happens.)

In detail, at any time t, $0 \leq t < T$, we obtain the conditions:

Upside constraints due to conversion:

$$\begin{aligned} &U(S,t) \geq \kappa S, \\ &B(S,t) = 0, \quad \text{if} \quad \hat{U} \leq \kappa S. \end{aligned}$$

Upside constraints due to callability by the CB issuer:

$$\begin{aligned} &U(S,t) \leq \max(B_c, \kappa S), \\ &B(S,t) = 0, \quad \text{if} \quad \hat{U} \geq B_c, \end{aligned}$$

where it is assumed that the holder has the right to convert if the issuer calls the CB.

Downside constraints due to puttability by the CB holder:

$$\begin{aligned} &U(S,t) \geq B_p, \\ &B(S,t) = B_p, \quad \text{if} \quad \hat{U} \leq B_p. \end{aligned}$$

To formulate the boundary conditions, we modify equations (2.7)-(2.8) at $S = 0$ and as $S \to \infty$. At $S = 0$, we have

$$\frac{\partial B}{\partial t} - (r + r_c)B \;=\; 0,$$

$$\frac{\partial U}{\partial t} - rU - r_c B = 0. \tag{2.11}$$

As $S \to \infty$ we assume that the unconstrained solution is linear in S, therefore

$$\begin{aligned} \frac{\partial^2 B}{\partial S^2} &= 0, \\ \frac{\partial^2 U}{\partial S^2} &= 0. \end{aligned} \tag{2.12}$$

Since the underlying stock price follows a lognormal random walk, using $x = \ln S$ we can convert Equations (2.7)-(2.8) into a simpler pair of diffusion equations:

$$\frac{\partial B}{\partial t} + \frac{\sigma^2}{2}\frac{\partial^2 B}{\partial x^2} + (r_g - \frac{\sigma^2}{2})\frac{\partial B}{\partial x} - (r + r_c)B + f(t) = 0. \tag{2.13}$$

$$\frac{\partial U}{\partial t} + \frac{\sigma^2}{2}\frac{\partial^2 U}{\partial x^2} + (r_g - \frac{\sigma^2}{2})\frac{\partial U}{\partial x} - r(U - B) - (r + r_c)B + f(t) = 0, \tag{2.14}$$

2.3 Model Derivation

We could not find a derivation of the TF model in the literature. Therefore, to understand the TF model, we derived the model through different approaches. We observe that in the TF model there are two rates, r and r_g, which are the risk-free interest rate and the stock growth rate, respectively. However, using r_g in equations (2.7) - (2.8) does not seem right to us. Neither the TF paper [16], nor the paper of Ayache, Forsyth, and Vetzal [1] explain why r_g should be included in these equations. In the example in the former paper, r is implicitly used for r_g; in the latter one, r is used for r_g. To understand these equations better, we derive them using three different approaches: the arbitrage-free approach, the probability of default approach, and the present value theory approach. The first approach is based on a discussion with Professor Robert Almgren, and uses the classical no-arbitrage theory of mathematical finance; the second approach is based on the probability perspective that is used in the paper of Ayache, Forsyth, and Vetzal [1]; the third one uses present value theory [2]. The first two approaches give rise to $r_g = r$, and the third one provides a possible explanation for using $r_g \neq r$. We tend to believe the arbitrage free approach is the most natural one in this context. In the later chapters, we use $r_g = r$ without further discussion.

Before discussing these approaches, we review some basic concepts. As is customary, we assume that the stock price is a stochastic process that follows the Geometric Brownian Motion

$$dS = \sigma S dW + \mu S dt, \tag{2.15}$$

where μ and σ are the drift rate and volatility, respectively, and W is a Wiener process. According to Itô's Lemma, any derivative $G = G(S,t)$ depending on S satisfies

$$dG = \left(\mu S \frac{\partial G}{\partial S} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 G}{\partial S^2} + \frac{\partial G}{\partial t}\right) dt + \sigma S \frac{\partial G}{\partial S} dW + o(dt), \tag{2.16}$$

assuming G is sufficiently smooth. The COCB price, B, and the CB price, U, are financial derivatives depending only on time and stock price (the risk-free interest rate is constant), so we have

$$dB = \sigma S \frac{\partial B}{\partial S} dW + \left(\mu S \frac{\partial B}{\partial S} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 B}{\partial S^2} + \frac{\partial B}{\partial t}\right) dt. \tag{2.17}$$

$$dU = \sigma S \frac{\partial U}{\partial S} dW + \left(\mu S \frac{\partial U}{\partial S} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 U}{\partial S^2} + \frac{\partial U}{\partial t}\right) dt, \tag{2.18}$$

These equations hold when there is no credit risk. Note that U includes B, while $U - B$ is the value that comes only from the convertibility.

2.3.1 Arbitrage Free Approach

In a short time interval, when there is no credit risk, the price change of a Convertible Bond would be the same as (2.18). If there is credit risk, it affects the price of the COCB part. The probability of loosing 100% of the value due to default at some time t can be "simulated" by a situation where a fraction of the money is lost per timestep. That is, during a small interval dt we assume we loose $r_c B dt$, where r_c is the credit spread. Therefore, we adjust (2.18) to obtain the following equation

$$dU = \sigma S \frac{\partial U}{\partial S} dW + \left(\mu S \frac{\partial U}{\partial S} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 U}{\partial S^2} + \frac{\partial U}{\partial t}\right) dt - r_c B dt. \tag{2.19}$$

Consider a portfolio Π that is long one CB and short Δ shares of stock: $\Pi = U - \Delta S$. Then

$$d\Pi = dU - \Delta dS, \tag{2.20}$$

Substituting (2.15) and (2.19) into (2.20), and choosing $\Delta = \frac{\partial U}{\partial S}$ to eliminate the stochastic term involving dW, we have

$$d\Pi = \left(\frac{\sigma^2 S^2}{2} \frac{\partial^2 U}{\partial S^2} + \frac{\partial U}{\partial t} - r_c B \right) dt. \tag{2.21}$$

To avoid creating an arbitrage opportunity, the portfolio must earn the risk-free interest rate, which implies that

$$d\Pi = r\Pi dt, \tag{2.22}$$

Thus

$$r\Pi dt = \left(\frac{\sigma^2 S^2}{2} \frac{\partial^2 U}{\partial S^2} + \frac{\partial U}{\partial t} - r_c B \right) dt, \tag{2.23}$$

hence

$$r\Pi = \frac{\sigma^2 S^2}{2} \frac{\partial^2 U}{\partial S^2} + \frac{\partial U}{\partial t} - r_c B. \tag{2.24}$$

Since $\Pi = U - S\frac{\partial U}{\partial S}$, we have

$$r(U - S\frac{\partial U}{\partial S}) = \frac{\sigma^2 S^2}{2} \frac{\partial^2 U}{\partial S^2} + \frac{\partial U}{\partial t} - r_c B, \tag{2.25}$$

which can be rewritten as

$$\frac{\partial U}{\partial t} + \frac{\sigma^2 S^2}{2} \frac{\partial^2 U}{\partial S^2} + rS\frac{\partial U}{\partial S} - rU - r_c B = 0. \tag{2.26}$$

This is the PDE (2.8) (excluding the cash flow f), since $-r(U - B) - (r + r_c)B = -rU - r_c B$. Similarly, we can obtain (2.7).

2.3.2 Probability of Default Approach

Consider again the portfolio of the last subsection,

$$d\Pi = dU - \Delta dS, \tag{2.27}$$

in the absence of default. Substituting (2.15) and (2.18) into (2.27), and similarly as before choosing $\Delta = \frac{\partial U}{\partial S}$, we get

$$d\Pi = \left[\frac{\partial U}{\partial t} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 U}{\partial S^2}\right] dt. \tag{2.28}$$

Now we consider default in a slightly different way than in the last subsection. Assume that

- the probability of default in the time interval $[t, t+dt]$ is pdt;
- on default, the bond holder will loose B (the whole COCB value, this is total default event);
- the stock price S is unchanged on default.

Then, equation (2.28) becomes

$$d\Pi = (1 - pdt)\left[\frac{\partial U}{\partial t} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 U}{\partial S^2}\right] dt - pdtB, \tag{2.29}$$

which becomes

$$d\Pi = \left[\frac{\partial U}{\partial t} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 U}{\partial S^2}\right] dt - pdtB, \tag{2.30}$$

where $\left[\frac{\partial U}{\partial t} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 U}{\partial S^2}\right] p(dt)^2$ is ignored when dt is small.

Assuming the default risk is diversifiable, the portfolio must earn risk free interest rate. Therefore,

$$E(d\Pi) = r\Pi dt = r(U - \Delta S)dt, \tag{2.31}$$

where $E(\cdot)$ denotes the expectation in the risk neutral world.

Combining (2.30) and (2.31), we have

$$\frac{\partial U}{\partial t} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 U}{\partial S^2} + rS\frac{\partial U}{\partial S} - rU - pB = 0. \tag{2.32}$$

Taking p to be the credit spread, we obtain the PDE (2.8). Similarly, we can obtain (2.7).

2.3.3 Present Value Theory Approach

At some future time $t+dt$, the value of COCB is $B(S+dS,t+dt)$. Suppose the discount rate for the COCB is ω, then

$$B(S,t) = \frac{1}{1+\omega dt} E(B(S+dS,t+dt)), \tag{2.33}$$

where $E(\cdot)$ denotes the expected value. The above equation can be rewritten as

$$\omega dt B(S,t) = E(B(S+dS,t+dt)) - B(S,t). \tag{2.34}$$

Since

$$E(B(S+dS,t+dt)) - B(S,t) = E(B(S,t)+dB) - B(S,t) = E(dB), \tag{2.35}$$

we have

$$\omega dt B(S,t) = E(dB). \tag{2.36}$$

According to Itô's Lemma,

$$dB = \sigma S \frac{\partial B}{\partial S} dW + \left(\mu S \frac{\partial B}{\partial S} + \frac{\sigma^2 S^2}{2} \frac{\partial^2 B}{\partial S^2} + \frac{\partial B}{\partial t} \right) dt. \tag{2.37}$$

Noting that

$$E(dW) = 0 \tag{2.38}$$

we have

$$E(dB) = \left(\mu S \frac{\partial B}{\partial S} + \frac{\sigma^2 S^2}{2} \frac{\partial^2 B}{\partial S^2} + \frac{\partial B}{\partial t} \right) dt. \tag{2.39}$$

Combining (2.36) and (2.39) gives

$$\frac{\partial B}{\partial t} + \frac{\sigma^2 S^2}{2} \frac{\partial^2 B}{\partial S^2} + \mu S \frac{\partial B}{\partial S} - \omega B = 0. \tag{2.40}$$

This is (2.7), if we regard ω as $(r+r_c)$. Since B is the risky bond component, this is reasonable because B should be discounted using the risky rate.

To derive (2.8), we follow similar steps, taking into account that, when discounting $U-B$, which is the equity component, the risk-free interest rate r should be used, and when discounting B, the risky rate $r+r_c$ should be used.

2.4 The Interpretation of Ayache, Forsyth, and Vetzal

Ayache, Forsyth, and Vetzal [1] extend the TF model. The authors explore the valuation of the Convertible Bond coupled with credit risk by providing detailed information about the handling of default: whether the stock price jumps to zero or not, how much the recovery rate is, etc. They argue that many of the existing models, such as TF [16], are incomplete, because there is no explicit specification about what happens in the event of a default by the issuing firm. In these models, it is implied that, upon a default, the firm's stock price does not change at all, or instantly jumps to zero; however, in reality, this is questionable in that the market must gradually react to the occurrence of default, and the stock price should not collapse suddenly. Therefore, an appropriate model should reflect that the firm's stock price falls to some value between 0% and 100% of the pre-default value.

This paper investigates a detailed problem solving procedure. It sets up a model without considering credit risk, analyzes the risky bond by taking into account the probability of default, and then adds credit risk into the model. The resulting model is referred to as partial default model. The TF model is a special case of the partial default model presented in [1], in that the stock price does not jump if default occurs. The CB is again split into the equity and COCB part, and the equations (2.7)-(2.8) still hold.

In order to solve the problem using the penalty method, the TF model is written as a linear complementarity problem. Let

$$\mathcal{N}(U,B) \equiv \frac{\partial U}{\partial t} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 U}{\partial S^2} + r_g S\frac{\partial U}{\partial S} - r(U-B) - (r+r_c)B. \tag{2.41}$$

Then the linear complementarity problem takes the form

- if $B_c > \kappa S$:

$$\begin{pmatrix} \mathcal{N}(U,B)=0 \\ (U-\max(B_p,\kappa S))\geq 0 \\ (U-B_c)\leq 0 \end{pmatrix}$$

$$\vee \begin{pmatrix} \mathcal{N}(U,B)\leq 0 \\ (U-\max(B_p,\kappa S))=0 \\ (U-B_c)\leq 0 \end{pmatrix}$$

$$\vee \begin{pmatrix} \mathcal{N}(U,B)\geq 0 \\ (U-\max(B_p,\kappa S))\geq 0 \\ (U-B_c)=0 \end{pmatrix};$$

- if $B_c \leq \kappa S$,

$$U=\kappa S.$$

Chapter 3

Discretization

To approximate the solution of the PDEs in Chapter 2, we use the Finite Difference Method (FDM) for both the spatial and temporal discretization. In this chapter, we provide the discretization for (2.8) for spatial grid points, and focus on three FDMs for the time discretization: the explicit, the fully implicit, and the Crank-Nicolson methods; then we interpret the constraints and boundary conditions, as well as the cash flows; finally we compare these methods in terms of stability and convergence.

We use the change of variables $\tau = T - t$ to transform the PDEs from forward time to backward time. Recall that V denotes the value of the American put option, U denotes that of the Convertible Bond, and B denotes that of the COCB part of the Convertible Bond. The Black-Scholes equation used for American option pricing becomes

$$\frac{\partial V}{\partial \tau} = \frac{\sigma^2 S^2}{2}\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV, \tag{3.1}$$

and the coupled PDEs for a Convertible Bond (2.7), (2.8) become

$$\frac{\partial B}{\partial \tau} = \frac{\sigma^2 S^2}{2}\frac{\partial^2 B}{\partial S^2} + rS\frac{\partial B}{\partial S} - (r + r_c)B, \tag{3.2}$$

$$\frac{\partial U}{\partial \tau} = \frac{\sigma^2 S^2}{2}\frac{\partial^2 U}{\partial S^2} + rS\frac{\partial U}{\partial S} - r(U - B) - (r + r_c)B. \tag{3.3}$$

We ignore $f(t)$ for now, leaving it for the discussion in Section 3.3.

Many researchers recommend the change of variables $x = \ln S$ for equations (3.1) to (3.3) to get

$$\frac{\partial V}{\partial \tau} = \frac{\sigma^2}{2}\frac{\partial^2 V}{\partial x^2} + (r - \frac{\sigma^2}{2})\frac{\partial V}{\partial x} - rV, \tag{3.4}$$
$$\frac{\partial B}{\partial \tau} = \frac{\sigma^2}{2}\frac{\partial^2 B}{\partial x^2} + (r - \frac{\sigma^2}{2})\frac{\partial B}{\partial x} - (r + r_c)B, \tag{3.5}$$
$$\frac{\partial U}{\partial \tau} = \frac{\sigma^2}{2}\frac{\partial^2 U}{\partial x^2} + (r - \frac{\sigma^2}{2})\frac{\partial U}{\partial x} - r(U - B) - (r + r_c)B. \tag{3.6}$$

We provide some results using this transformation in Chapter 5, but in the rest of the chapters, we develop the methods for the original equations (3.1) to (3.3).

We choose a set of grid points in forward time

$$\{t_0, t_1, t_2, \ldots, t_N\}, \qquad t_0 = 0 < t_1 < \cdots < t_N = T. \tag{3.7}$$

Define $\Delta t_n = t_{n+1} - t_n$, $n = 0, 1, \ldots, N-1$. Usually we choose

$$\Delta t_0 = \Delta t_1 = \cdots = \Delta t_{N-1} = \Delta t, \tag{3.8}$$

but we may choose nonuniform timesteps to improve efficiency.

We use τ_n to denote the backward time points: $\tau_n = t_{N-n}, n = 0, \ldots, N$. Suppose we choose N timesteps with uniform stepsize $\Delta\tau = T/N$. Then the time grid points for backward computing would be

$$\tau_n = T - n\Delta\tau, \quad n = 0, 1, \ldots, N.$$

Define $S_i \equiv i\Delta S$, $i = 0, 1, \ldots, m$, to be the uniform spatial grid points, where ΔS is the grid spacing, and note that S_0 and S_m are boundary points. Define $U_i^n \approx U(S_i, \tau_n)$ to be the approximation to the solution of (3.3) at asset value S_i and time τ_n. Let U^n denote the vector $(U_0^n\ U_1^n\ \cdots\ U_m^n)^T$ at time τ_n. In the following, we investigate the rule for computing U^{n+1} from U^n, and then present similar formulae for B and V.

3.1 Finite Differences

A finite difference (FD) provides an approximation to a derivative. The first time-derivative of U is often approximated by

$$\frac{\partial U}{\partial \tau}(S_i, \tau_n) \approx \frac{U_i^n - U_i^{n-1}}{\Delta \tau}. \tag{3.9}$$

This is called a backward difference. The corresponding forward difference is

$$\frac{\partial U}{\partial \tau}(S_i, \tau_n) \approx \frac{U_i^{n+1} - U_i^n}{\Delta \tau}. \tag{3.10}$$

The first spatial-derivative $\frac{\partial U}{\partial S}(S_i, \tau_n)$ has three common approximations:

$$\frac{U_{i+1}^n - U_i^n}{\Delta S}, \qquad \frac{U_i^n - U_{i-1}^n}{\Delta S}, \qquad \text{and} \qquad \frac{U_{i+1}^n - U_{i-1}^n}{2\Delta S}. \tag{3.11}$$

These are called the forward difference, the backward difference, and the central difference, respectively.

One of these approximations is asymptotically more accurate than the others. From a Taylor series expansion of $U(S + \Delta S, \tau)$ about the point (S, τ) we have

$$U(S + \Delta S, \tau) = U(S, \tau) + \Delta S \frac{\partial U}{\partial S}(S, \tau) + \frac{\Delta S^2}{2} \frac{\partial^2 U}{\partial S^2}(S, \tau) + O(\Delta S^3). \tag{3.12}$$

Similarly,

$$U(S - \Delta S, \tau) = U(S, \tau) - \Delta S \frac{\partial U}{\partial S}(S, \tau) + \frac{\Delta S^2}{2} \frac{\partial^2 U}{\partial S^2}(S, \tau) + O(\Delta S^3). \tag{3.13}$$

Subtracting one from the other, dividing by $2\Delta S$ and rearranging gives

$$\frac{\partial U}{\partial S}(S, \tau) = \frac{U_{i+1}^n - U_{i-1}^n}{2\Delta S} + O(\Delta S^2). \tag{3.14}$$

Thus the central difference has a truncation error (the error in the approximation of the derivative) of $O(\Delta S^2)$, whereas the truncation error for the forward and backward differences are $O(\Delta S)$. Consequently, the central difference is more accurate asymptotically than the forward or backward difference because of the fortunate cancellation of terms, due to the symmetry about S in the definition of the central difference. We use the central difference as the approximation to the first spatial-derivative throughout this thesis.

The second derivative with respect to S is frequently approximated by the central difference formula

$$\frac{\partial^2 U}{\partial S^2}(S_i, \tau_n) \approx \frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{\Delta S^2}. \tag{3.15}$$

Again, this comes from a Taylor series expansion. It is easy to show, as we did in (3.14), that the truncation error in this approximation is also $O(\Delta S^2)$.

3.2 Finite Difference Methods

Consider (3.3) and a uniform spatial discretization of $[0, \infty]$. Clearly, $S_0 = 0$. Since numerically we cannot set $S_m = \infty$, we use a large number L to approximate ∞ and set $S_m = L$. We will discuss how to choose this large number in Chapter 5. Consider also a parameter θ, called the *implicitness* parameter, that satisfies $0 \le \theta \le 1$. Then using the FD approximations from the previous section, the differential equation (3.3) can be approximated by the finite difference equation

$$\begin{aligned} \frac{U_i^{n+1} - U_i^n}{\Delta \tau} = \; & (1-\theta)\left(\frac{\sigma^2 S_i^2}{2}\frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{\Delta S^2} + rS_i\frac{U_{i+1}^n - U_{i-1}^n}{2\Delta S} - rU_i^n\right) \\ & +\theta\left(\frac{\sigma^2 S_i^2}{2}\frac{U_{i+1}^{n+1} - 2U_i^{n+1} + U_{i-1}^{n+1}}{\Delta S^2} + rS_i\frac{U_{i+1}^{n+1} - U_{i-1}^{n+1}}{2\Delta S} - rU_i^{n+1}\right) \\ & -r_c B_i^{n+1}, \end{aligned} \tag{3.16}$$

for $i = 1, \ldots, m-1$.

We refer to this as the θ-method, although we use $r_c B_i^{n+1}$ in (3.16) instead of $(1-\theta) r_c B_i^n + \theta r_c B_i^{n+1}$ that would appear in the standard θ-method. We found that the standard θ-method gives poor results; our modified one in (3.16) provides better results. The intuitive reason, we believe, is that the current value of U_i should be dependent on the most recent value of B_i.

For $i = 1, \ldots, m-1$, define

$$\begin{aligned} \alpha_i &= (\frac{\sigma^2 S_i^2}{2\Delta S^2} - \frac{rS_i}{2\Delta S})\Delta\tau, \qquad (3.17) \\ \beta_i &= (\frac{\sigma^2 S_i^2}{2\Delta S^2} + \frac{rS_i}{2\Delta S})\Delta\tau. \end{aligned}$$

From now on, whenever it is clear from the context, we will drop the indication of the i index range.

Then (3.16) becomes

$$\begin{aligned} U_i^{n+1} - U_i^n &= (1-\theta)\left(\alpha_i U_{i-1}^n - (r\Delta\tau + \alpha_i + \beta_i)U_i^n + \beta_i U_{i+1}^n\right) \\ &+\theta\left(\alpha_i U_{i-1}^{n+1} - (r\Delta\tau + \alpha_i + \beta_i)U_i^{n+1} + \beta_i U_{i+1}^{n+1}\right) \\ &-r_c\Delta\tau B_i^{n+1}. \qquad (3.18) \end{aligned}$$

Moving all the terms involving U^{n+1} in (3.18) to the left, and the terms involving U^n to the right, we obtain

$$\begin{aligned} & U_i^{n+1} - \theta\left(\alpha_i U_{i-1}^{n+1} - (r\Delta\tau + \alpha_i + \beta_i)U_i^{n+1} + \beta_i U_{i+1}^{n+1}\right) \\ = \; & U_i^n + (1-\theta)\left(\alpha_i U_{i-1}^n - (r\Delta\tau + \alpha_i + \beta_i)U_i^n + \beta_i U_{i+1}^n\right) - r_c\Delta\tau B_i^{n+1}. \qquad (3.19) \end{aligned}$$

Similarly, for B and V, we have

$$\begin{aligned} & B_i^{n+1} - \theta\left(\alpha_i B_{i-1}^{n+1} - ((r+r_c)\Delta\tau + \alpha_i + \beta_i)B_i^{n+1} + \beta_i B_{i+1}^{n+1}\right) \\ = \; & B_i^n + (1-\theta)\left(\alpha_i B_{i-1}^n - ((r+r_c)\Delta\tau + \alpha_i + \beta_i)B_i^n + \beta_i B_{i+1}^n\right), \qquad (3.20) \end{aligned}$$

and

$$\begin{aligned} & V_i^{n+1} - \theta\left(\alpha_i V_{i-1}^{n+1} - (r\Delta\tau + \alpha_i + \beta_i)V_i^{n+1} + \beta_i V_{i+1}^{n+1}\right) \\ = \; & V_i^n + (1-\theta)\left(\alpha_i V_{i-1}^n - (r\Delta\tau + \alpha_i + \beta_i)V_i^n + \beta_i V_{i+1}^n\right). \qquad (3.21) \end{aligned}$$

Note that we have used uniform grid points in the whole spatial domain. This wastes points in regions where there is little variability of the solution. We could use non-uniform grid points and put more grid points in regions where the solution changes rapidly and fewer grid points where the solution changes more slowly. The finite differences would have to be adjusted accordingly.

Notice also that B_i^{n+1} must be computed before U_i^{n+1}, since the (3.19) involves B_i^{n+1}. This does not mean that B_i^{n+1} is independent of U_i^{n+1}. As explained in section 3.3, the free (interior) boundary conditions introduce dependency between B_i^{n+1} and U_i^{n+1}.

We next discuss three specific finite difference methods (FDMs) for discretizing the time variable in the PDEs in (3.1)-(3.3), namely the explicit method, the Crank-Nicolson method, and the fully implicit method. These methods arise when θ in (3.19)-(3.21) is set to certain values. Specifically,

when	$\theta = 0$,	we get the explicit or Forward Euler method;
when	$\theta = 1/2$,	we get the Crank-Nicolson Method or Trapezoidal Rule;
when	$\theta = 1$,	we get the fully implicit or Backward Euler method.

3.2.1 Explicit FDM

In (3.19), when $\theta = 0$, we have

$$U_i^{n+1} = \alpha_i U_{i-1}^n + (1 - (r\Delta\tau + \alpha_i + \beta_i))U_i^n + \beta_i U_{i+1}^n - r_c \Delta\tau B_i^{n+1}. \tag{3.22}$$

In the above equation, U_i^{n+1} is given by a difference formula in terms of $U_{i-1}^n, U_i^n, U_{i+1}^n$, and B_i^{n+1}. The time derivative uses the option values at times τ_n and τ_{n+1}, whereas all other terms use values at τ_n, given that B_i^{n+1} has been calculated before hand. Because the value of U_i^{n+1} can be computed explicitly from the already computed values U_{i-1}^n, U_i^n, and U_{i+1}^n at time τ_n, this method is called the *explicit* finite difference method. The truncation error for the explicit FDM is $O(\Delta t + \Delta S^2)$.
Similarly, for B and V, we have

$$B_i^{n+1} = \alpha_i B_{i-1}^n + (1 - ((r + r_c)\Delta\tau + \alpha_i + \beta_i))B_i^n + \beta_i B_{i+1}^n, \tag{3.23}$$

and

$$V_i^{n+1} = \alpha_i V_{i-1}^n + (1 - (r\Delta\tau + \alpha_i + \beta_i))V_i^n + \beta_i V_{i+1}^n. \tag{3.24}$$

The (3.23) and (3.24) are traditional explicit methods, whereas (3.22) is not quite a traditional explicit method because of the B_i^{n+1} term.

3.2.2 Fully Implicit FDM

In (3.19), when $\theta = 1$, we have

$$-\alpha_i U_{i-1}^{n+1} + (1 + r\Delta\tau + \alpha_i + \beta_i)U_i^{n+1} - \beta_i U_{i+1}^{n+1} = U_i^n - r_c \Delta\tau B_i^{n+1}. \qquad (3.25)$$

In the above equation, U_i^{n+1} is related by a difference formula with U_i^n, U_{i-1}^{n+1}, U_{i+1}^{n+1} and B_i^{n+1}. Consequently, U_i^{n+1} cannot be explicitly computed in terms of only past values of U (i.e., values at timestep n). This method is called the *fully implicit* finite difference method. The truncation error is $O(\Delta t + \Delta S^2)$.

Similarly, for B and V, we have

$$-\alpha_i B_{i-1}^{n+1} + (1 + (r + r_c)\Delta\tau + \alpha_i + \beta_i)B_i^{n+1} - \beta_i B_{i+1}^{n+1} = B_i^n, \qquad (3.26)$$

and

$$-\alpha_i V_{i-1}^{n+1} + (1 + r\Delta\tau + \alpha_i + \beta_i)V_i^{n+1} - \beta_i V_{i+1}^{n+1} = V_i^n. \qquad (3.27)$$

3.2.3 Crank-Nicolson Method

The *Crank-Nicolson* method can be thought of as an average of the explicit method and the fully implicit method. It uses six U values, three at time τ_n and three at time τ_{n+1}. In (3.19), when $\theta = \frac{1}{2}$, we have the Crank-Nicolson scheme

$$\begin{aligned} & -\alpha_i U_{i-1}^{n+1} + (2 + r\Delta\tau + \alpha_i + \beta_i)U_i^{n+1} - \beta_i U_{i+1}^{n+1} \\ = \; & \alpha_i U_{i-1}^n + (2 - (r\Delta\tau + \alpha_i + \beta_i))U_i^n + \beta_i U_{i+1}^n - r_c \Delta\tau B_i^{n+1}. \end{aligned} \qquad (3.28)$$

The truncation error for the traditional Crank-Nicolson method is $O(\Delta t^2 + \Delta S^2)$, but, because of the term of $-r_c \Delta\tau B_i^{n+1}$, in our modified version (3.28) of this method, the truncation error is $O(\Delta t^\alpha + \Delta S^2)$, where $1 < \alpha < 2$. Hence the truncation error for U in our modified version is not as small as that in the traditional Crank-Nicolson method. Similarly, for B and V, we have

$$\begin{aligned} & -\alpha_i B_{i-1}^{n+1} + (2 + (r + r_c)\Delta\tau + \alpha_i + \beta_i)B_i^{n+1} - \beta_i B_{i+1}^{n+1} \\ = \; & \alpha_i B_{i-1}^n + (2 - ((r + r_c)\Delta\tau + \alpha_i + \beta_i))B_i^n + \beta_i B_{i+1}^n, \end{aligned} \tag{3.29}$$

and

$$\begin{aligned} & -\alpha_i V_{i-1}^{n+1} + (2 + r\Delta\tau + \alpha_i + \beta_i)V_i^{n+1} - \beta_i V_{i+1}^{n+1} \\ = \; & \alpha_i V_{i-1}^n + (2 - (r\Delta\tau + \alpha_i + \beta_i))V_i^n + \beta_i V_{i+1}^n. \end{aligned} \tag{3.30}$$

Note that (3.29) and (3.30) are implementations of the traditional Crank-Nicolson method. We have not modified them.

3.3 Boundary Conditions, Coupon Payment, and Accrued Interest

Now we discuss how to incorporate the boundary conditions, the coupon payments, and the accrued interest.

When $S = 0$, (3.2) and (3.3) reduce to (2.11), which can be discretized as

$$\begin{aligned} \frac{B_0^{n+1} - B_0^n}{\Delta\tau} &= -\theta(r + r_c)B_0^{n+1} - (1 - \theta)(r + r_c)B_0^n, \\ \frac{U_0^{n+1} - U_0^n}{\Delta\tau} &= -\theta r U_0^{n+1} - (1 - \theta)rU_0^n - r_c B_0^{n+1}, \end{aligned}$$

and for the American option, this boundary condition is

$$\frac{V_0^{n+1} - V_0^n}{\Delta\tau} = -\theta r V_0^{n+1} - (1 - \theta)rV_0^n. \tag{3.31}$$

The above can be rearranged to

$$\begin{aligned} (1 + \theta(r + r_c)\Delta\tau)B_0^{n+1} &= (1 - (1 - \theta)(r + r_c)\Delta\tau)B_0^n, \\ (1 + \theta r\Delta\tau)U_0^{n+1} &= (1 - (1 - \theta)r\Delta\tau)U_0^n - r_c\Delta\tau B_0^{n+1}, \end{aligned} \tag{3.32}$$

and

$$(1+\theta r\Delta\tau)V_0^{n+1} = (1-(1-\theta)r\Delta\tau)V_0^n. \tag{3.33}$$

For large values of S, the finite difference representation for the boundary condition (2.12) is

$$\begin{aligned} B_m^{n+1} &= 2B_{m-1}^{n+1} - B_{m-2}^{n+1}, \\ U_m^{n+1} &= 2U_{m-1}^{n+1} - U_{m-2}^{n+1}. \end{aligned} \tag{3.34}$$

For the American option, it is

$$V_m^{n+1} = 2V_{m-1}^{n+1} - V_{m-2}^{n+1}. \tag{3.35}$$

An alternative boundary condition for a Convertible Bond for large S is

$$B_m^{n+1} = 0, \qquad U_m^{n+1} = \kappa S_m. \tag{3.36}$$

And for an American put option, it is

$$V_m^{n+1} = 0. \tag{3.37}$$

In each iteration of PSOR method, we use (3.34) and (3.35); in the penalty method, we use (3.36) and (3.37).

We now discuss the handling of coupon payments given by (2.9). Let t_n^+ be the forward time the instant after a coupon payment, and t_n^- the forward time the instant before a coupon payment. Then the discrete coupon payments are handled by setting

$$\begin{aligned} U(S,t_n^-) &= U(S,t_n^+) + c_n, \\ B(S,t_n^-) &= B(S,t_n^+) + c_n. \end{aligned} \tag{3.38}$$

The payoff condition for the Convertible Bond at $t = T$ is

$$U(S,T) = \max(\kappa S, F + c_{last}), \tag{3.39}$$

where c_{last} is the last coupon payment at T. For semiannual coupon payments equal to C, assuming T (the number of years to the maturity of the bond) is an integer, we have $c_{last} = C$.

Let t be the current time in the forward direction, t_k^c the time of the previous coupon payment, and t_{k+1}^c the time of the next pending coupon payment. Then, between two coupon payments, we have $t_k^c \leq t < t_{k+1}^c$. The accrued interest on the pending coupon payment at time t is

$$AccI(t) = c_{k+1} \frac{t - t_k^c}{t_{k+1}^c - t_k^c}. \tag{3.40}$$

Usually quoted prices are clean prices; dirty prices include any accrued interest that has accumulated since the last coupon payment. The dirty call price B_c and the dirty put price B_p are given by

$$\begin{aligned} B_c(t) &= B_c^{cl} + AccI(t), \\ B_p(t) &= B_p^{cl} + AccI(t), \end{aligned} \tag{3.41}$$

where B_c^{cl} and B_p^{cl} are the respective clean prices.

For programming purposes, in order that the $B_c(t)$ and the $B_p(t)$ work globally regardless of the period of callability and puttability, we assign special values to $B_c(t)$ and the $B_p(t)$. In the period when the Convertible Bond is not callable, we assign $B_c(t)$ a large value so that the call event is unlikely to happen; in the period when the Convertible Bond is not puttable, we assign zero to $B_p(t)$ so that the put event never happens.

According to Hull [10], we suppose that the convertibility, and the exercise of the call and put would happen immediately after the coupon payment; and in our numerical procedure, we apply the call, put, and converting constraints first, and then add the coupon payment (because we are proceeding backwards in time).

3.4 Matrix Formulation

In this section, we describe how the discretized equations can be written in matrix format. Recall (3.19), and define the $(m+1) \times (m+1)$ matrix $\mathbf{M_U}$ by

$$\mathbf{M_U} = \begin{pmatrix} -r\Delta\tau & 0 & 0 & \dots & 0 \\ \alpha_1 & -(r\Delta\tau + \alpha_1 + \beta_1) & \beta_1 & \dots & 0 \\ 0 & \alpha_2 & -(r\Delta\tau + \alpha_2 + \beta_2) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \beta_{m-1} \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

The matrix $\mathbf{M_U}$ is tridiagonal, that is, except for three central diagonals all the elements of the matrix are zero.

Recall that

$$U^n = (U_0^n \;\; U_1^n \;\; \cdots \;\; U_{m-1}^n \;\; U_m^n)^T, \tag{3.42}$$

and

$$B^n = (B_0^n \;\; B_1^n \;\; \cdots \;\; B_{m-1}^n \;\; B_m^n)^T. \tag{3.43}$$

Let $\mathbf{I}$ be the identity matrix of size $(m+1) \times (m+1)$. The equation (3.19), for $i = 1, \dots, m-1$, together with the boundary conditions for U in (3.32) and (3.36) can be written in matrix format as

$$(\mathbf{I} - \theta\mathbf{M_U})U^{n+1} = (\mathbf{I} + (1-\theta)\mathbf{M_U})U^n - r_c\Delta\tau B^{n+1}. \tag{3.44}$$

Similarly, define

$$\mathbf{M_B} = \begin{pmatrix} -(r+r_c)\Delta\tau & 0 & 0 & \dots & 0 \\ \alpha_1 & -((r+r_c)\Delta\tau + \alpha_1 + \beta_1) & \beta_1 & \dots & 0 \\ 0 & \alpha_2 & -((r+r_c)\Delta\tau + \alpha_2 + \beta_2) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \beta_{m-1} \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

and the matrix formulation of equations (3.20), for $i = 1, \ldots, m-1$, together with the boundary conditions for B in (3.32) and (3.36) is

$$(\mathbf{I} - \theta \mathbf{M_B})B^{n+1} = (\mathbf{I} + (1-\theta)\mathbf{M_B})B^n. \tag{3.45}$$

For an American option, the matrix formula is similar to (3.45) with the respective matrix being $\mathbf{M} = \mathbf{M_U}$. Recall that

$$V^n = (V_0^n \;\; V_1^n \;\; \cdots \;\; V_{m-1}^n \;\; V_m^n)^T, \tag{3.46}$$

so we have

$$(\mathbf{I} - \theta \mathbf{M})V^{n+1} = (\mathbf{I} + (1-\theta)\mathbf{M})V^n. \tag{3.47}$$

Note that the matrix formulations (3.44) and (3.45) include the alternative boundary conditions of (3.36) for large S. More specifically, the calculation of the last row in (3.44) results in

$$U_m^{n+1} = U_m^n, \tag{3.48}$$

and eventually, for all the timesteps, it becomes

$$U_m^{n+1} = U_m^n = U_m^n = \cdots = U_m^0. \tag{3.49}$$

Since $U_m^0 = \kappa S_m$, the alternative boundary conditions for U in (3.36) have been included in (3.44). Similarly, the calculation of the last row in (3.45) results in

$$B_m^{n+1} = B_m^n, \tag{3.50}$$

and eventually, for all the timesteps, it becomes

$$B_m^{n+1} = B_m^n = B_m^n = \cdots = B_m^0, \tag{3.51}$$

which, since $B_m^0 = 0$, implements the alternative boundary conditions for B in (3.36).

Note that the formulations (3.44), (3.45), and (3.47) are *linear problems.* However, this does not mean that a linear system is solved at each time step. When the free (interior) boundary conditions are taken into account, eventually a *non-linear problem* results at each time step. These non-linear problems are solved by iterative methods as discussed in the next chapter. Note also that, if we prefer to apply the boundary conditions (3.34) and (3.35) at the right end, we cannot incorporate them into the matrix formulation. However, at each timestep, we can easily adjust the values of B_m^{n+1} and U_m^{n+1} according to (3.34), and the value of V_m^{n+1} according to (3.35), after having solved the problems arising from (3.44), (3.45), and (3.47).

Note that for $\theta \neq 0$, the matrices on the left side of (3.44) to (3.47) are tridiagonal. In the case of $\theta = 0$, i.e., the case of the explicit method, the matrices on the left side of (3.44) to (3.47) degenerate to the identity, therefore no system needs to be solved. This is, of course, the property of any explicit method.

3.5 Stability and Convergence

The explicit method is conditionally stable and convergent; the time and spatial stepsizes must satisfy a relationship of the form Δt and ΔS to preserve stability. The relationship is usually $\Delta t \leq a\Delta S^2$, where a is a constant. The fully implicit method and the Crank-Nicolson method are unconditionally stable and convergent; there is no restriction of the above form on Δt and ΔS. The explicit method and the fully implicit method are both first order methods with respect to the timestep Δt and second order with respect to ΔS. That is, the truncation error is $O(\Delta t + \Delta S^2)$. An advantage of the fully implicit method is that it smooths discontinuities in the initial conditions and, as noted above, does not have a stability restriction on the choice of the timestep Δt. The standard Crank-Nicolson method is a second order method with respect to both Δt and ΔS. Its truncation error is $O(\Delta t^2 + \Delta S^2)$. Thus, it has a higher convergence rate with respect to Δt than the previous two methods. However, our modified Crank-Nicolson method (3.28) has a truncation error that is not as small as $O(\Delta t^2 + \Delta S^2)$.

Chapter 4

Iterative Methods

To solve a linear system, we can use either a direct or an iterative method. A direct method attempts to solve the problem in one step; an iterative method starts with an initial guess, and successively improves it until it is sufficiently close to the exact solution. However, as noted in the last chapter, though relations (3.44), (3.45), and (3.47) are linear systems, we cannot just solve them and proceed to the next time step. We need to take into account the free (interior) boundary conditions, and eventually solve a non-linear problem. A direct method cannot incorporate the free boundary conditions arising from callability, puttability, and conversion; while an iterative method, when appropriately adjusted, can. The Projected Successive Over-Relaxation (PSOR) method and the penalty method used with an implicit Finite Difference method are iterative methods appropriate for handling the free boundary conditions. In this chapter, we discuss both the PSOR and the penalty methods which are used to solve the non-linear problems arising from (3.44), (3.45), and (3.47), when the free boundary conditions are taken into account.

4.1 PSOR Method

Before we consider PSOR, we review the Successive Overrelaxation (SOR) method, an iterative method for solving systems of linear algebraic equations. SOR is an extension of the Gauss-Seidel method, and can be derived from it by introducing a relaxation parameter. The computation of the i^{th} vector component by SOR takes the form of a weighted average between the previous iterate and the computed Gauss-Seidel iterate,

$$\widetilde{V}_i^{n,k} = V_i^{n,k-1} + \omega(\overline{V}_i^{n,k} - V_i^{n,k-1}), \tag{4.1}$$

where k is the iteration index, n the timestep index, $\overline{V}^{n,k}$ is the respective Gauss-Seidel iterate, $V^{n,k-1}$ the previous iterate, and ω the relaxation factor, where $0 < \omega < 2$. When $\omega = 1$, formula (4.1) reduces to the Gauss-Seidel method; when $1 < \omega < 2$, it is called over-correction or over-relaxation; and when $0 < \omega < 1$, it is called under-relaxation. However, because in many cases the optimal relaxation factor satisfies $1 < \omega < 2$, the term over-relaxation is used generically.

The PSOR method is an extension of the SOR method for solving free boundary value problems such as those that arise for American options. We consider PSOR for American options first for simplicity.

For American options, every iteration involves comparing the value of the option that would be obtained if you don't exercise the option to the payoff value $V_i^{n+1,*}$ that you would obtain if you do exercise the option, and taking the larger of these two values as the value of the option at that point. That is, the PSOR iteration for an American option at node S_i and time τ_{n+1} is

$$V_i^{n+1,k} = \max(V_i^{n+1,*}, \widetilde{V}_i^{n+1,k}), \tag{4.2}$$

where $V_i^{n+1,*}$ is the payoff at the τ_{n+1}, and

$$\widetilde{V}_i^{n+1,k} = V_i^{n+1,k-1} + \omega(\overline{V}_i^{n+1,k} - V_i^{n+1,k-1}). \tag{4.3}$$

Consider (3.30), the Crank-Nicolson formula for an American option. One Gauss-Seidel iterate $\overline{V}_i^{n+1,k}$ is computed by

$$\begin{aligned}\overline{V}_i^{n+1,k} = & \frac{1}{2+r\Delta\tau+\alpha_i+\beta_i}\Big(\alpha_i V_{i-1}^{n+1,k}+\beta_i V_{i+1}^{n+1,k-1} \\ & +\alpha_i V_{i-1}^n + (2-(r\Delta\tau+\alpha_i+\beta_i))V_i^n+\beta_i V_{i+1}^n\Big). \end{aligned} \tag{4.4}$$

For a put option, the payoff is

$$V_i^{n+1,*} = \max(0, E - S_i), \tag{4.5}$$

where E is the exercise price. For a call option, the payoff function would be different.

For a Convertible Bond, we calculate B then U. The fully implicit method in (3.26) gives

$$(1+(r+r_c)\Delta\tau+\alpha_i+\beta_i)B_i^{n+1} = \alpha_i B_{i-1}^{n+1}+\beta_i B_{i+1}^{n+1}+B_i^n, \tag{4.6}$$

and (3.25) gives

$$(1+r\Delta\tau+\alpha_i+\beta_i)U_i^{n+1} = \alpha_i U_{i-1}^{n+1}+\beta_i U_{i+1}^{n+1}+U_i^n - r_c\Delta\tau B_i^{n+1}. \tag{4.7}$$

So the Gauss-Seidel iterates are

$$\begin{aligned}\overline{B}_i^{n+1,k} &= \frac{1}{1+(r+r_c)\Delta\tau+\alpha_i+\beta_i}(\alpha_i B_{i-1}^{n+1,k}+\beta_i B_{i+1}^{n+1,k-1}+B_i^n),\\ \overline{U}_i^{n+1,k} &= \frac{1}{1+r\Delta\tau+\alpha_i+\beta_i}(\alpha_i U_{i-1}^{n+1,k}+\beta_i U_{i+1}^{n+1,k-1}+U_i^n - r_c\Delta\tau B_i^{n+1,k}),\end{aligned} \tag{4.8}$$

and the SOR values of B_i^{n+1} and U_i^{n+1} at iteration k are given by

$$\widetilde{B}_i^{n+1,k} = B_i^{n+1,k-1} + \omega(\overline{B}_i^{n+1,k} - B_i^{n+1,k-1}), \tag{4.9}$$

$$\widetilde{U}_i^{n+1,k} = U_i^{n+1,k-1} + \omega(\overline{U}_i^{n+1,k} - U_i^{n+1,k-1}). \tag{4.10}$$

Now we apply all the convertible constraints to $\widetilde{U}_i^{n+1,k}$ and $\widetilde{B}_i^{n+1,k}$.

First, check the callability if the bond is callable at this time:

$$\text{if} \quad \widetilde{U}_i^{n+1,k} > \max(B_c(\tau_{n+1}), \kappa S_i) \quad \text{then} \quad \begin{aligned} U_i^{n+1,k} &= \max(B_c(\tau_{n+1}), \kappa S_i),\\ B_i^{n+1,k} &= 0.\end{aligned}$$

Next check the puttability if the bond is puttable at this time:

$$\text{if} \quad \widetilde{U}_i^{n+1,k} < B_p(\tau_{n+1}) \quad \text{and} \quad \kappa S_i < B_p(\tau_{n+1}) \quad \text{then} \qquad \begin{aligned} U_i^{n+1,k} &= B_p(\tau_{n+1}), \\ B_i^{n+1,k} &= B_p(\tau_{n+1}). \end{aligned}$$

At last, check if it is more profitable to convert if the bond is convertible at this time:

$$\text{if} \quad \widetilde{U}_i^{n+1,k} < \kappa S_i \quad \text{and} \quad B_p(\tau_{n+1}) < \kappa S_i \quad \text{then} \qquad \begin{aligned} U_i^{n+1,k} &= \kappa S_i, \\ B_i^{n+1,k} &= 0. \end{aligned}$$

The final condition for exiting the iteration is

$$\begin{aligned} ||B_i^{n+1,k} - B_i^{n+1,k-1}|| &\leq \epsilon, \\ ||U_i^{n+1,k} - U_i^{n+1,k-1}|| &\leq \epsilon, \end{aligned} \tag{4.11}$$

where ϵ is the tolerance. When this condition is satisfied, we set

$$\begin{aligned} U_i^{n+1} &= U_i^{n+1,k}, \\ B_i^{n+1} &= B_i^{n+1,k}. \end{aligned} \tag{4.12}$$

For the Crank-Nicolson method PSOR is similar except that

$$\begin{aligned} \overline{B}_i^{n+1,k} = \quad & \frac{1}{2 + (r + r_c)\Delta\tau + \alpha_i + \beta_i} \Big(\alpha_i B_{i-1}^{n+1} + \beta_i B_{i+1}^{n+1} \\ & + \alpha_i B_{i-1}^n + (2 - ((r + r_c)\Delta\tau + \alpha_i + \beta_i)) B_i^n + \beta_i B_{i+1}^n \Big). \end{aligned} \tag{4.13}$$

and

$$\begin{aligned} \overline{U}_i^{n+1,k} = \quad & \frac{1}{2 + r\Delta\tau + \alpha_i + \beta_i} \Big(\alpha_i U_{i-1}^{n+1} + \beta_i U_{i+1}^{n+1} \\ & + \alpha_i U_{i-1}^n + (2 - (r\Delta\tau + \alpha_i + \beta_i)) U_i^n + \beta_i U_{i+1}^n - r_c \Delta\tau B_i^{n+1} \Big). \end{aligned} \tag{4.14}$$

The pseudocode for the Crank-Nicolson timstepping and PSOR iteration are given in Algorithm 1 and Algorithm 2.

Algorithm 1 Crank-Nicolson timestepping for a Convertible Bond

$U^0 = B^0 = F + C$; $\{F$: face value of the bond; C: coupon payment at maturity$\}$

for $i = 0$ to m **do**

  if $U_i^0 < \kappa S_i$ **then**

    $U_i^0 = \kappa S_i$; $B_i^0 = 0$;

  end if

end for

for $n = 0$ to $N - 1$ **do**

  $\tau = T - (n+1)\Delta\tau$;$\{$ T is the maturity, $\Delta\tau$ is the time stepsize$\}$

  calculate AccI using (3.40);

  if $\tau \in$ call period **then**

    $B_c = B_c^{cl} + AccI$;

  else

    $B_c = L$; $\{$assigning a big number to B_c in the non-callable period$\}$

  end if

  if $\tau \in$ put period **then**

    $B_p = B_p^{cl} + AccI$;

  else

    $B_p = 0$; $\{$assigning 0 to B_p in the non-puttable period$\}$

  end if

  calculate U_0^{n+1} and B_0^{n+1} using (3.32);

  call PSOR iteration;

  calculate U_m^{n+1} and B_m^{n+1} using (3.34);

  if $\tau \in$ coupon payment period **then**

    $U^{n+1} = U^{n+1} + C$; $B^{n+1} = B^{n+1} + C$;

  end if

end for

Algorithm 2 PSOR iteration for a Convertible Bond

for $k = 1$ to MAXLOOP **do**

$error_u = error_b = 0;$

for $i = 1$ to $m - 1$ **do**

calculate $\overline{B}_i^{n+1,k}$ using (4.13);

calculate $\widetilde{B}_i^{n+1,k}$ using (4.9); $B_i^{n+1,k} = \widetilde{B}_i^{n+1,k}$

calculate $\overline{U}_i^{n+1,k}$ using (4.14);

calculate $\widetilde{U}_i^{n+1,k}$ using (4.10); $U_i^{n+1,k} = \widetilde{U}_i^{n+1,k}$

if $U_i^{n+1,k} > \max(B_c, \kappa S_i)$ **then**

$U_i^{n+1,k} = \max(B_c, \kappa S_i);$

$B_i^{n+1,k} = 0;$

end if

if $U_i^{n+1,k} < B_p$ **then**

$U_i^{n+1,k} = B_p;$

$B_i^{n+1,k} = B_p;$

end if

if $U_i^{n+1,k} < \kappa S_i$ **then**

$U_i^{n+1,k} = \kappa S_i;$

$B_i^{n+1,k} = 0;$

end if

$error_u = error_u + (U_i^{n+1,k-1} - U_i^{n+1,k})^2;$

$error_b = error_b + (B_i^{n+1,k-1} - B_i^{n+1,k})^2;$

end for

if $error_u \leq \epsilon^2$ and $error_b \leq \epsilon^2$ **then**

break;

end if

end for

$U^{n+1} = U^{n+1,k};\ B^{n+1} = B^{n+1,k};$

4.2 Penalty Method

The PSOR method explicitly applies the constraints to the linear complementarity problem, whereas the penalty method applies the constraints implicitly, using a nonsmooth Newton iteration. We discuss the penalty scheme for an American option first, and then extend it to a Convertible Bond.

4.2.1 Penalty Scheme

As described in Chapter 2, the complementarity problem for an American put option is

$$\begin{aligned} \mathcal{L}V &\leq 0, \\ (V - V^*) &\geq 0, \\ (\mathcal{L}V = 0) &\vee (V - V^* = 0), \end{aligned} \tag{4.15}$$

where V^* is the payoff function, the notation $(\mathcal{L}V = 0) \vee (V - V^* = 0)$ means that either $(\mathcal{L}V = 0)$ or $(V - V^* = 0)$ at each point in the solution domain, and recall that

$$\mathcal{L}V \equiv -\frac{\partial V}{\partial \tau} + (\frac{\sigma^2 S^2}{2}\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV). \tag{4.16}$$

The payoff for an American put option is

$$V^*(S) = \max(E - S, 0). \tag{4.17}$$

The penalty scheme is based on the nonlinear PDE

$$\frac{\partial V}{\partial \tau} = \frac{\sigma^2 S^2}{2}\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV + \rho \max(V^* - V, 0), \tag{4.18}$$

where the solution satisfies $V \geq V^*$, and, in the limit as the positive penalty parameter $\rho \to \infty$, V approaches V^*.

4.2.2 Penalty Discretization for American Options

Except for the penalty part, the discretization of equation (4.18) has been discussed in the previous chapter. By denoting the penalty term by q_i^{n+1}, we can write the discretization scheme as

$$\mathcal{F}V_i^{n+1} = q_i^{n+1}, \tag{4.19}$$

where

$$\begin{aligned}\mathcal{F}V_i^{n+1} &= (V_i^{n+1} - V_i^n) \\ &\quad -\theta\left(\alpha_i V_{i-1}^{n+1} - (r\Delta\tau + \alpha_i + \beta_i)V_i^{n+1} + \beta_i V_{i+1}^{n+1}\right) \\ &\quad -(1-\theta)\left(\alpha_i V_{i-1}^{n} - (r\Delta\tau + \alpha_i + \beta_i)V_i^{n} + \beta_i V_{i+1}^{n}\right),\end{aligned} \tag{4.20}$$

and

$$q_i^{n+1} = \begin{cases} (V_i^* - V_i^{n+1})Large & \text{if } V_i^{n+1} < V_i^*, \\ 0 & \text{otherwise,}\end{cases}$$

where *Large* is a large positive number, called the penalty factor.

The whole discretization scheme for the American option is

$$\begin{aligned}V_i^{n+1} - V_i^n &= \theta\left(\alpha_i V_{i-1}^{n+1} - (r\Delta\tau + \alpha_i + \beta_i)V_i^{n+1} + \beta_i V_{i+1}^{n+1}\right) \\ &\quad +(1-\theta)\left(\alpha_i V_{i-1}^{n} - (r\Delta\tau + \alpha_i + \beta_i)V_i^{n} + \beta_i V_{i+1}^{n}\right) \\ &\quad +P(V_i^{n+1})(V_i^* - V_i^{n+1})\end{aligned} \tag{4.21}$$

where

$$P(V_i^{n+1}) = \begin{cases} Large & \text{if } V_i^{n+1} < V_i^*, \\ 0 & \text{otherwise.}\end{cases}$$

To write (4.21) in matrix form, let the diagonal matrix $\mathbf{P}(V^{n+1})$ of size $(m+1)\times(m+1)$ be given by

$$\mathbf{P}(V^{n+1})_{ij} = \begin{cases} Large & \text{if } V_i^{n+1} < V_i^* \text{ and } i = j, \\ 0 & \text{otherwise.}\end{cases}$$

Then the matrix form of (4.21) is

$$[\mathbf{I} - \theta\mathbf{M} + \mathbf{P}(V^{n+1})]V^{n+1} = (\mathbf{I} + (1-\theta)\mathbf{M})V^n + [\mathbf{P}(V^{n+1})]V^*. \tag{4.22}$$

It has been proved (see [1]) that the solution of (4.22) satisfies

$$V_i^{n+1} - V_i^* \to 0 \quad \text{as} \quad Large \to \infty \tag{4.23}$$

for nodes where $\mathcal{F}V_i^{n+1} > 0$. If we require that the Linear Complementarity Problem (LCP) be computed with a relative precision of tol for those nodes where $V_i^{n+1} < V_i^*$, we should have $Large \simeq 1/tol$.

In theory, if we are taking the limit as ΔS, $\Delta\tau \to 0$, then for the Crank-Nicolson method, we should have

$$Large = O(\frac{1}{\min[(\Delta S)^2, (\Delta\tau)^2]}). \tag{4.24}$$

However, the method used for the Convertible Bond pricing is not the standard Crank-Nicolson method, and a reasonable choice for $Large$ is

$$Large = O(\frac{1}{\min[(\Delta S)^2, (\Delta\tau)^\alpha]}), \tag{4.25}$$

where $1 < \alpha < 2$. The above formula means that any error in the penalized formulation would tend to zero at the same rate as the discretization error. However, in practice, we can specify the value of $Large$ in terms of the required accuracy. In other words, we specify the maximum allowed error in the discrete penalized problem. We reduce ΔS, $\Delta\tau$ until the discretization error is reduced to this level of accuracy.

4.2.3 Penalty Discretization for Convertible Bonds

Convertible Bonds are much more complicated than American options. The whole Convertible Bond value U depends on the value of the bond component B, which follows a process similar to that of U. On one hand, B should be evaluated separately before the evaluation of U; on the other hand, B has to be adjusted after U has been evaluated to incorporate the convertibility, puttability and callability constraints.

Because the convertibility is through the whole life of the Convertible Bond, the payoff function U^* is

$$\begin{aligned} &\text{when} \quad B_c > \kappa S, && U^* = \max(B_p, \kappa S), \\ &\text{when} \quad B_c \le \kappa S, && U^* = \kappa S. \end{aligned} \tag{4.26}$$

The penalty discretization scheme for the Convertible Bond is

$$
\begin{aligned}
U_i^{n+1} - U_i^n &= \theta\left(\alpha_i U_{i-1}^{n+1} - (r\Delta\tau + \alpha_i + \beta_i)U_i^{n+1} + \beta_i U_{i+1}^{n+1}\right) \\
&\quad +(1-\theta)\left(\alpha_i U_{i-1}^{n} - (r\Delta\tau + \alpha_i + \beta_i)U_i^{n} + \beta_i U_{i+1}^{n}\right) \\
&\quad -r_c\Delta\tau B_i^{n+1} \\
&\quad +P(U_i^{n+1})(U_i^* - U_i^{n+1}), \qquad (4.27)
\end{aligned}
$$

where

$$
P(U_i^{n+1}) = \begin{cases} Large & \text{if } U_i^{n+1} < U_i^*, \\ 0 & \text{otherwise.} \end{cases}
$$

To write (4.27) in matrix form, let the diagonal matrix $\mathbf{P}(U^{n+1})$ of size $(m+1)\times(m+1)$ be given by

$$
\mathbf{P}(U^{n+1})_{ij} = \begin{cases} Large & \text{if } U_i^{n+1} < U_i^* \text{ and } i = j, \\ 0 & \text{otherwise.} \end{cases}
$$

Then the matrix form of (4.27) is

$$
[\mathbf{I} - \theta\mathbf{M_U} + \mathbf{P}(U^{n+1})]U^{n+1} = (\mathbf{I} + (1-\theta)\mathbf{M_U})U^n - r_c\Delta\tau B^{n+1} + [\mathbf{P}(U^{n+1})]U^*. \quad (4.28)
$$

4.2.4 Penalty Iteration

Let $V^{n+1,k}$ be the kth estimate for V^{n+1}. We briefly describe the penalty iteration for an American option in Algorithm 3.

Algorithm 3 Brief description of penalty iteration on American options

for $k = 0, ...,$ until convergence **do**

  solve $[\mathbf{I} - \theta\mathbf{M} + \mathbf{P}(V^{n+1,k}]V^{n+1,k+1} = (\mathbf{I} + (1-\theta)\mathbf{M})V^n + [\mathbf{P}(V^{n+1,k})]V^*$;

  if $\left[\max_i \frac{|V_i^{n+1,k+1} - V_i^{n+1,k}|}{\max(1, |V_i^{n+1,k+1}|)} < tol\right]$ or $[\mathbf{P}(V^{n+1,k+1}) = \mathbf{P}(V^{n+1,k})]$ **then**

    exit from for loop;

  end if

end for

The penalty iteration for a Convertible Bond is different from that of an American option due to the Convertible Bond's complexity. First, we estimate B^{n+1} by ignoring any constraints; then this value is used to compute U^{n+1} using the implicit method; next, we check the constraints explicitly; at last, we use a few steps of the penalty iteration to enhance the convergence. The timestepping algorithm and the penalty iteration on a Convertible Bond are shown in Algorithms 4 and 5.

4.2.5 About the Penalty Method

The penalty method has the advantage of finite termination (in exact arithmetic). For an iterate sufficiently close to the solution, the algorithm terminates in one iteration. This is especially advantageous when dealing with American option pricing, since, for each penalty iteration, we have an excellent initial guess from the previous timestep. In fact, as we shall see, for typical grids and timesteps, the algorithm takes, on average, less than two iterations per timestep to converge. Finite termination also implies that the number of iterations required for convergence is insensitive to the size of the penalty factor (until the limit of machine precision is reached). Another advantage is that the iteration is globally convergent using full Newton steps.

The disadvantage of the penalty method is that the constraints are satisfied only approximately, but since this error can be easily made to be much smaller than the discretization error, this does not appear to be a practical disadvantage.

Algorithm 4 Penalty timestepping for a Convertible Bond

$U = B = F + C$; $\{F$: face value of the bond; C: coupon payment at maturity$\}$

for $i = 0$ to m **do**

  if $U_i < \kappa S_i$ **then**

    $U_i = \kappa S_i$; $B_i = 0$;

  end if

end for

for $n = 0$ to $N - 1$ **do**

  $\tau = T - (n + 1)\Delta\tau$;$\{T$ is the maturity, $\Delta\tau$ is the time interval $\}$

  calculate AccI using (3.40);

  if $\tau \in$ call period **then**

    $B_c = B_c^{cl} + AccI$;

  else

    $B_c = L$; $\{L$ is the number representing $\infty\}$

  end if

  if $\tau \in$ put period **then**

    $B_p = B_p^{cl} + AccI$;

  else

    $B_p = 0$;

  end if

  calculate B^{n+1} using implicit method in (3.45);

  calculate U^{n+1} using B^{n+1} and implicit method in (3.44);

  apply constraints explicitly, similarly as in the PSOR iteration;

  call Penalty iteration with Crank-Nicolson method for 3 iterations;

  $B^{n+1} = \min(B^{n+1}, U^{n+1})$;

  if $\tau \in$ coupon payment period **then**

    $U^{n+1} = U^{n+1} + C$; $B^{n+1} = B^{n+1} + C$;

  end if

end for

Algorithm 5 Penalty iteration for a Convertible Bond

$U^* = \max(B_p, \kappa S)$;

for $i = 0$ to m **do**

  if $U_i^{n+1,0} < U_i^*$ **then**

    $P(U_i^{n+1,0}) = Large$;

  else

    $P(U_i^{n+1,0}) = 0$;

  end if

end for

for $k = 0, ...,$ until convergence **do**

  solve (4.28);

  for $i = 0$ to m **do**

    if $B_c \leq \kappa S_i$ **then**

      $U_i^{n+1,k+1} = \kappa S_i$;

    end if

    if $U_i^{n+1,k+1} \leq \kappa S_i$ **then**

      $P(U_i^{n+1,k+1}) = Large$;

    else

      $P(U_i^{n+1,k+1}) = 0$;

    end if

  end for

  if $\left[\max_i \frac{|U_i^{n+1,k+1} - U_i^{n+1,k}|}{\max(1, |U_i^{n+1,k+1}|)} < tol\right]$ or $[P(U_i^{n+1,k+1}) = P(U_i^{n+1,k})]$ **then**

    exit from for loop;

  end if

end for

$U^{n+1} = U^{n+1,k+1}$;

Chapter 5

Numerical Results

In this chapter, we illustrate results from the methods described in Chapter 4, starting with the American option pricing, followed by the more complex Convertible Bond pricing. In order to study the Convertible Bond, we investigate both the simple Convertible Bond and the full-featured Convertible Bond. The simple Convertible Bond is only convertible, and has no coupon, put or call features. The full-featured Convertible Bond includes all the following features: coupon payment, convertibility, callability, and puttability. In this chapter, we first discuss replacing the boundary condition at $S = \infty$ in the continuous model by a boundary condition at a finite value of S in the finite difference method. For comparison purposes, we include the results from the explicit method as well as from implicit methods. For the implicit methods, we include a version of each method that uses the PSOR iteration and another version that uses the penalty iteration. Finally we present some plots based on these results.

Our first numerical example is an American option with parameter values shown in Table 5.1. The parameters for the Convertible Bond are shown in Table 5.2. In the numerical result tables, "Price" is the at-the-money price, which is 100 in our examples; "Nodes" is the number of spatial grid points; "Difference" or "Diff" is the absolute value of the change in the solution as the grid is refined; "Ratio" is the ratio of successive differences; and "Time" is the computation time. Unless otherwise specified, either the timestep is halved at each grid refinement, or the spatial grid is halved at each timestep level.

Table 5.1: Model parameters for the American put option

Parameter	Value
Time to expiry T	0.25 years
Interest rate r	10% or 0.10
Exercise price E	100
Volatility σ	80% or 0.80
Tolerance ϵ	1.0e-07

Table 5.2: Model parameters for the Convertible Bond

Parameter	Value
Time to expiry T	5 years
Conversion	0 to 5 years into 1 share
Conversion ratio κ	1.0
Clean call price B_c^{cl}	110 from year 2 to year 5
Clean put price B_p^{cl}	105 at 3 years (during the third year)
Coupon payments	$4.0, semiannually
Coupon dates	.5, 1.0, 1.5, ..., 5.0
Risk-free interest rate r	5% or 0.05
Credit risk r_c	2% or 0.02
Volatility σ	20% or 0.20

5.1 Replacing the Boundary Condition at Infinity by a Finite Boundary Condition

If the asset-stepsize ΔS and the time-stepsize $\Delta\tau$ are constant, the grid is made up of the points at asset values

$$S = i\Delta S, \quad i = 0, 1, \ldots, I, \tag{5.1}$$

and times

$$\tau = T - n\Delta\tau, \quad n = 0, 1, \ldots, N. \tag{5.2}$$

That is, the option price is computed at asset values from zero up to the asset value $S = I\Delta S$. Noting that the Black-Scholes equation is to be solved for $0 \leq S < \infty$, we are effectively replacing an infinite spatial domain in the continuous problem by a finite domain in the finite difference approximation. In a sense, $I\Delta S$ is our approximation to infinity. In practice, this upper limit does not have to be too large: usually three or four times the value of the exercise price, or more generally, three or four times the largest asset value at which we are interested in the option price [17] suffices to get satisfactory results.

In our examples, the American option has the exercise price of 100, so we choose 500 as the upper limit for the asset values; the Convertible Bond has a put price of 105, a call price of 110, and the principal of 100, therefore 500 is also a reasonable upper limit on the asset value. To show that 500 is a reasonable upper limit on the asset value in the finite difference method, two numerical experimental results are presented in Tables 5.3 and 5.4 with different upper limits.

The results in Tables 5.3 and 5.4 are for an American put option with parameter values given in Table 5.1. We use the PSOR iteration associated with the Crank-Nicolson method to solve for the free boundary. Table 5.3 shows the numerical results with the upper limit of $S = 500$; Table 5.4 shows the numerical results with the upper limit of $S = 1000$. From these two tables, we observe

Table 5.3: American put option price with an upper limit of $S = 500$

TimeSteps	Nodes	Price	Time
100	25	14.16785097	0s
200	50	14.54745925	0s
400	100	14.64501132	2s
800	200	14.67024054	6s
1600	400	14.67668766	24s
3200	800	14.67832267	95s
6400	1600	14.67873688	390s
12800	3200	14.67883348	1562s

Table 5.4: American put option price with an upper limit of $S = 1000$

TimeSteps	Nodes	Price	Time
100	50	14.16785097	0s
200	100	14.54745925	1s
400	200	14.64501132	3s
800	400	14.67024054	14s
1600	800	14.67668767	54s
3200	1600	14.67832268	216s
6400	3200	14.67873298	925s
12800	6400	14.67883214	4080s

- At coarse grids, the respective prices in the two tables are exactly the same to (at least) 8 digits after the decimal dot; therefore the difference in the boundary condition is negligible compared to the discretization error.

- At finest grids, the difference in values with the same $\Delta\tau$ and ΔS but different boundary condition is smaller than the difference between results with same boundary condition but different $\Delta\tau$ and ΔS. Moreover, the numerical results appear to have converged to 5 decimal places. Hence the effect of the different boundary is negligible.

- In terms of the execution time, at the same timestep level, the experiment in Table 5.3 handles fewer grid points, and thus takes less time than the experiment in Table 5.4. For example, at 12800 timesteps level, the former one takes 1526 seconds, while the latter one takes 4080 seconds, which is about 2.5 times of the former one. Therefore, using the upper bound of 1000 does not help in getting substantially better results; in fact, it takes much longer time than when using the upper bound of 500.

Hence, for pricing purposes it is enough to take three or four times of the at-the-money asset value as the upper limit. All the numerical results in the rest of this chapter are based on this upper limit.

When using the original variable S, it is easy and natural to use 0 as the left boundary. However, when using the transformed formulae with $S = Ee^x$ or $S = e^x$, the left boundary cannot be exactly 0, since $S = 0$ corresponds to $x = -\infty$. For this reason, when the transformed formulae are used, we let the left boundary be a small positive number Ee^{x_L} or $e^{\tilde{x}_L}$, for some values x_L or $\tilde{x}_L$ sufficiently small, so that $Ee^{x_L} \approx 0$ or $e^{\tilde{x}_L} \approx 0$, which implies that $V(Ee^{x_L}) \approx V(0)$ or $V(e^{\tilde{x}_L}) \approx V(0)$, respectively.

The finite boundary conditions also apply to the Convertible Bond pricing.

5.2 Explicit Method Results

The explicit method is conditionally convergent. It converges if and only if the time and space stepsizes satisfy a relationship of the form

$$\Delta\tau \leq a\Delta S^2 \qquad \text{or} \qquad \Delta\tau \leq \tilde{a}\Delta x^2 \tag{5.3}$$

as $\Delta\tau$, ΔS, and Δx tend to zero, where a and $\tilde{a}$ are constants that depend on the volatility parameter σ. In this section, numerical results are provided for an American put option as well as a Convertible Bond with both the original variable S and the transformed variable x. The experimental results for a and $\tilde{a}$ for the appropriate relationships between $\Delta\tau$ and ΔS^2, as well as $\Delta\tau$ and Δx^2 are also included.

Table 5.5 shows some numerical results under different relationships between $\Delta\tau$ and ΔS^2 for pricing an American put option using the explicit method with variable S. The trial includes several different values of a. Using $a = 10^{-5}$ produces meaningless price values, while using $a = 10^{-6}$ and $a = 10^{-7}$ give reasonable results. This indicates that $a = 10^{-5}$ is inappropriate, and the relationship of $\Delta\tau < 10^{-5}\Delta S^2$ is necessary.

Table 5.5: Results of explicit method for an American put option with variable S ($\Delta\tau = a\Delta S^2$)

	$a = 10^{-5}$		$a = 10^{-6}$		$a = 10^{-7}$	
TimeSteps	Nodes	Price	Nodes	Price	Nodes	Price
1000	100	∞	32	13.5007	10	11.0840
2000	141	$-\infty$	45	14.5114	14	15.0186
4000	200	NaN	63	14.5223	20	13.8407
8000	283	NaN	89	14.6317	28	14.9754

The convergence results when $\Delta\tau = 10^{-7}\Delta S^2$ are shown in Table 5.6. The average ratio is about 4.0, which is consistent with the theory that the explicit method is first order with respect to $\Delta\tau$, and second order with respect to ΔS, as stated in Chapter 4.

Table 5.6: Results of explicit method for an American put option ($\Delta\tau = 10^{-7}\Delta S^2$)

TimeSteps($\Delta\tau$)	Nodes(ΔS)	Price	Diff	Ratio
1000(2.500000e-04)	10(5.000000e+01)	11.08400480		
4000(6.250000e-05)	20(2.500000e+01)	13.84071902	2.75671422	
16000(1.562500e-05)	40(1.250000e+01)	14.47433604	0.63361702	4.4
64000(3.906250e-06)	80(6.250000e+00)	14.62659548	0.15225944	4.2
256000(9.765625e-07)	160(3.125000e+00)	14.66548984	0.03889436	3.9
1024000(2.441406e-07)	320(1.562500e+00)	14.67548647	0.00999663	3.9
4096000(6.103516e-08)	640(7.812500e-01)	14.67802416	0.00253768	3.9
16384000(1.525879e-08)	1280(3.906250e-01)	14.67866398	0.00063982	4.0

As to the algorithm that uses transformation to the x variable, both $S = Ee^x$ and $S = e^x$ are implemented. In the first case, using $S = Ee^x$, we are able to have $x = 0$ for $S = E$ or $S = 100$ as a spatial point, and thus interpolation is not needed. The numerical results with different $\tilde{a}$ are shown in Table 5.7.

In Table 5.7, the interval $[-16, 2]$ is used as the x domain, which is about $[0, 739]$ when mapping to the asset S domain. From this table, we observe that $\tilde{a} = 1.0$ and $\tilde{a} = 1.5$ lead to convergent results; however, $\tilde{a} = 2.0$ produces invalid results. Therefore, the value of $\tilde{a}$ should be smaller than 2, which means $\Delta\tau < 2\Delta x^2$. The convergence ratio is theoretically consistent with the first order method.

In the second case, using the transformation $S = e^x$, since the volatility σ is not changed, the relationship between $\Delta\tau$ and Δx^2 should also follow $\Delta\tau < 2\Delta x^2$. The value $e^6 \approx 403$ is used as the upper limit, $e^{-10} \approx 0$ as the lower limit, and $\tilde{a} = 1$. The numerical results are shown in Table 5.8.

The ratio in Table 5.8 is unstable and oscillates. A possible reason for the instability

Table 5.7: Results of explicit method for an American put option with $S = Ee^x$ ($\Delta\tau = \tilde{a}\Delta x^2$)

$\tilde{a}$	TimeSteps($\Delta\tau$)	Nodes(Δx)	Price	Difference	Ratio
	1(2.500000e-01)	37(5.000000e-01)	14.75510026		
	4(6.250000e-02)	73(2.500000e-01)	14.55989614	0.19520412	
	16(1.562500e-02)	145(1.250000e-01)	14.64755463	0.08765848	2.2
	64(3.906250e-03)	289(6.250000e-02)	14.67161274	0.02405812	3.6
1.00	256(9.765625e-04)	577(3.125000e-02)	14.67722778	0.00561504	4.3
	1024(2.441406e-04)	1153(1.562500e-02)	14.67848968	0.00126190	4.4
	4096(6.103516e-05)	2305(7.812500e-03)	14.67878578	0.00029610	4.3
	16384(1.525879e-05)	4609(3.906250e-03)	14.67885605	0.00007028	4.2
	65536(3.814697e-06)	9217(1.953125e-03)	14.67887291	0.00001686	4.2
	10(2.343750e-02)	145(1.250000e-01)	14.18639051		
	42(5.859375e-03)	289(6.250000e-02)	14.59038979	0.40399928	
1.50	170(1.464844e-03)	577(3.125000e-02)	14.65756419	0.06717440	6.0
	682(3.662109e-04)	1153(1.562500e-02)	14.67358074	0.01601655	4.2
	2730(9.155273e-05)	2305(7.812500e-03)	14.67755797	0.00397722	4.0
	10922(2.288818e-05)	4609(3.906250e-03)	14.67854899	0.00099102	4.0
	8(3.125000e-02)	145(1.250000e-01)	0.71412233		
2.00	32(7.812500e-03)	289(6.250000e-02)	0.00000000	0.71412233	
	128(1.953125e-03)	577(3.125000e-02)	0.00000000	0.71412233	0.0

Table 5.8: Results of explicit method for an American put option with $S = e^x$ ($\Delta\tau = \Delta x^2$)

TimeSteps($\Delta\tau$)	Nodes(Δx)	Price	Difference	Ratio
16(1.562500e-02)	129(1.250000e-01)	14.82052987		
64(3.906250e-03)	257(6.250000e-02)	14.74719495	0.07333492	
256(9.765625e-04)	513(3.125000e-02)	14.69798816	0.04920679	1.5
1024(2.441406e-04)	1025(1.562500e-02)	14.68283283	0.01515533	3.2
4096(6.103516e-05)	2049(7.812500e-03)	14.68017481	0.00265802	5.7
16384(1.525879e-05)	4097(3.906250e-03)	14.67895295	0.00122187	2.2
65536(3.814697e-06)	8193(1.953125e-03)	14.67891791	0.00003503	34.9

and oscillation of the ratio is the interpolation error, since the spatial point $S = 100$ could not be used as a grid point.

In theory, the convergence for the explicit method is first order with respect to Δt. The American option follows this rule even though in some cases there are oscillations. However, the Convertible Bond is different. Tables 5.9, 5.10, and 5.11 show the numerical results for the full-featured Convertible Bond with the original variable S, the transformation $S = 100e^x$, and the transformation $S = e^x$, respectively.

Table 5.9 compares the convergence using the original variable S under different relationships between $\Delta\tau$ and ΔS^2. Note that the domain of S is from 0 to 500. From this table, we observe that the relationship must satisfy $\Delta\tau < 5 \times 10^{-4}\Delta S^2$ in order to get a convergent result. The convergence ratio oscillates for both $a = 5 \times 10^{-5}$ and $a = 10^{-4}$.

Table 5.10 compares the convergence using different relationships between $\Delta\tau$ and Δx^2 under the transformation $S = 100e^x$. Note the price in this table does not need interpolation. Note also that the asset value boundary is $100e^{-16}$ to $100e^2$, corresponding to 0 to 739.

Table 5.9: Results of explicit method for a full-featured CB with variable S ($\Delta\tau = a\Delta S^2$)

a	TimeSteps($\Delta\tau$)	Nodes(ΔS)	Price	Difference	Ratio
5×10^{-5}	4000(0.001250)	100(5.000000)	124.0284121014		
	16000(0.000313)	200(2.500000)	124.0053413340	0.02307077	
	64000(0.000078)	400(1.250000)	123.9839671353	0.02137420	1.1
	256000(0.000020)	800(0.625000)	123.9758623279	0.00810481	2.6
	1024000(0.000005)	1600(0.312500)	123.9668138635	0.00904846	0.9
	4096000(0.000001)	3200(0.156250)	123.9657495979	0.00106427	8.5
10^{-4}	2000(0.002500)	100(5.000000)	124.0289946695		
	8000(0.000625)	200(2.500000)	124.0055282520	0.02346642	
	32000(0.000156)	400(1.250000)	123.9840247217	0.02150353	1.1
	128000(0.000039)	800(0.625000)	123.9758753760	0.00814935	2.6
	512000(0.000010)	1600(0.312500)	123.9668173846	0.00905799	0.9
	2048000(0.000002)	3200(0.156250)	123.9657504119	0.00106697	8.5
5×10^{-4}	400(0.012500)	100(5.000000)	163.8637564689		
	1600(0.003125)	200(2.500000)	1348.2945075105	1184.43075104	
	6400(0.000781)	400(1.250000)	2478.0481364818	1129.75362897	1.0

Table 5.10: Results of explicit method for a full-featured CB with $S = 100e^x$ ($\Delta\tau = \tilde{a}\Delta x^2$)

$\tilde{a}$	TimeSteps($\Delta\tau$)	Nodes(Δx)	Price	Difference	Ratio
4	5(1.000000)	36(0.500000)	110.1092533972		
	20(0.250000)	72(0.250000)	124.7527972258	14.64354383	
	80(0.062500)	144(0.125000)	124.2182078334	0.53458939	27.4
	320(0.015625)	288(0.062500)	124.1103153917	0.10789244	5.0
	1280(0.003906)	576(0.031250)	124.0055769958	0.10473840	1.0
	5120(0.000977)	1152(0.015625)	123.9886676390	0.01690936	6.2
	20480(0.000244)	2304(0.007812)	123.9734474118	0.01522023	1.1
16	5(1.000000)	72(0.250000)	112.2273641107		
	20(0.250000)	144(0.125000)	124.3931848005	12.16582069	
	80(0.062500)	288(0.062500)	124.1398153881	0.25336941	48.0
	320(0.015625)	576(0.031250)	124.0132501233	0.12656526	2.0
	1280(0.003906)	1152(0.015625)	123.9906885045	0.02256162	5.6
	5120(0.000977)	2304(0.007812)	123.9739021053	0.01678640	1.3
25	3(1.666667)	72(0.250000)	111.5362689282		
	13(0.384615)	144(0.125000)	123.9991362700	12.46286734	
	51(0.098039)	288(0.062500)	124.1816042654	0.18246800	68.3
	205(0.024390)	576(0.031250)	123.6543044680	0.52729980	0.3
	819(0.006105)	1152(0.015625)	123.7638903385	0.10958587	4.8
	3277(0.001526)	2304(0.007812)	123.7235920937	0.04029824	2.7
	13107(0.000381)	4608(0.003906)	123.7423945749	0.01880248	2.1

In Table 5.10, we observe that when $\tilde{a} = 25$, the price itself oscillates, and goes far away from the correct price 123.97 [1], while, when $\tilde{a} = 16$ and $\tilde{a} = 4$, the price converges to the correct one. Therefore, the relationship between $\Delta\tau$ and Δx^2 for the Convertible Bond should be $\Delta\tau < 25\Delta x^2$. The ratio oscillates for both $\tilde{a} = 4$ and $\tilde{a} = 16$. Table 5.11 shows a similar oscillation when $S = e^x$ is used with the asset boundary limits of e^{-10} and e^6, and $\tilde{a} = 16$.

Table 5.11: Results of explicit method for a full-featured CB with $S = e^x$ ($\Delta\tau = 16\Delta x^2$)

TimeSteps($\Delta\tau$)	Nodes(Δx)	Price	Difference	Ratio
5(1.000000)	64(0.250000)	113.7250412746		
20(0.250000)	128(0.125000)	124.6058332351	10.88079196	
80(0.062500)	256(0.062500)	124.1904104671	0.41542277	26.2
320(0.015625)	512(0.031250)	124.0413151474	0.14909532	2.8
1280(0.003906)	1024(0.015625)	123.9913983384	0.04991681	3.0
5120(0.000977)	2048(0.007812)	123.9827058205	0.00869252	5.7
20480(0.000244)	4096(0.003906)	123.9737963056	0.00890951	1.0
81920(0.000061)	8192(0.001953)	123.9687492124	0.00504709	1.8

All the numerical results above reveal that the explicit method is not efficient for pricing Convertible Bonds. Hence, implicit methods combined with iterative techniques are used to improve efficiency.

5.3 Implicit Method Results

Both the Crank-Nicolson and the fully implicit methods require iterative methods to handle the free boundary problems. We study two iterative methods, the PSOR and the penalty methods. In our implementation, unless specifically stated, we always combine one of the above iterative methods with the Crank-Nicolson time discretization method. While using the iterative approach, we focus on comparing the number of iterations. In the results, "max" is the maximum number of iterations over all timesteps; "min" is the minimum number of iterations; "avg" is the average number of iterations; and "Iters" is the total number of iterations used in all timesteps. The tolerance is $eps = 10^{-6}$, unless otherwise specified.

5.3.1 PSOR Results

Tables 5.12, 5.13 and 5.14 show the numerical results for an American put option, a simple Convertible Bond and a full-featured Convertible Bond, respectively, using the PSOR method.

The results of Table 5.12 are obtained using the S variable. In Table 5.12, the convergence ratio for the American option is stable, with the average of 4.0, and a standard deviation of 0.16. This is consistent with a second order method. Assuming the correct answer is 14.6788 [6], the PSOR iteration obtained this result at timestep level of 1.95×10^{-5}; while in Table 5.6, where the S variable is also used, the timestep in the explicit method must go smaller than 1.53×10^{-8} to reach this result. Thus the PSOR iteration requires substantially fewer timesteps than the explicit method to reach the solution. Even taking into account that each PSOR timestep is more expensive than the explicit method timestep, since the PSOR timestep consists of about 10 iterations, the PSOR method is much more efficient than the explicit method.

However, the Convertible Bond shows lower convergence ratio, as shown in Tables

Table 5.12: Results of PSOR method for an American put option

TimeSteps($\Delta\tau$)	Nodes(ΔS)	Price	Diff	Ratio	No. of Iterations max	min	avg
100(2.500000e-03)	26(2.0000e+01)	14.16785097			11	4	5.1
200(1.250000e-03)	51(1.0000e+01)	14.54745925	0.37960828		11	4	5.3
400(6.250000e-04)	101(5.0000e+00)	14.64501132	0.09755208	3.9	11	4	5.5
800(3.125000e-04)	201(2.5000e+00)	14.67024054	0.02522922	3.9	11	4	6.1
1600(1.562500e-04)	401(1.2500e+00)	14.67668766	0.00644712	3.9	11	4	6.9
3200(7.812500e-05)	801(6.2500e-01)	14.67832267	0.00163501	3.9	11	4	7.8
6400(3.906250e-05)	1601(3.1250e-01)	14.67873688	0.00041421	3.9	13	5	8.9
12800(1.953125e-05)	3201(1.5625e-01)	14.67883348	0.00009660	4.3	19	5	11.5

5.13 and 5.14.

Table 5.13: Results of PSOR method for a simple CB with $S = 100e^x$

TimeSteps($\Delta\tau$)	Nodes(Δx)	Price	Diff	Ratio	No. of Iterations max	min	avg
100(0.050000)	240(0.075000)	104.02124615			9	5	6.5
200(0.025000)	480(0.037500)	104.16175781	0.14051166		11	5	6.4
400(0.012500)	960(0.018750)	104.22601957	0.06426176	2.2	13	5	6.7
800(0.006250)	1920(0.009375)	104.25671597	0.03069640	2.1	19	7	7.7
1600(0.003125)	3840(0.004688)	104.27171203	0.01499606	2.0	30	8	9.6
3200(0.001563)	7680(0.002344)	104.27912277	0.00741074	2.0	50	10	13.2
6400(0.000781)	15360(0.001172)	104.28280677	0.00368399	2.0	90	15	16.3
12800(0.000391)	30720(0.000586)	104.28468766	0.00188089	2.0	165	18	22.5

Table 5.14: Results of PSOR method for a full-featured CB with $S = 100e^x$

TimeSteps($\Delta\tau$)	Nodes(Δx)	Price	Diff	Ratio	No. of Iterations		
					max	min	avg
100(0.050000)	240(0.075000)	124.11539278			10	7	8.1
200(0.025000)	480(0.037500)	124.05237745	0.06301533		14	8	9.2
400(0.012500)	960(0.018750)	123.99817763	0.05419981	1.2	17	9	11.0
800(0.006250)	1920(0.009375)	123.98226223	0.01591540	3.4	23	12	14.0
1600(0.003125)	3840(0.004688)	123.97327463	0.00898759	1.8	30	15	17.8
3200(0.001563)	7680(0.002344)	123.96977707	0.00349757	2.6	51	18	23.3
6400(0.000781)	15360(0.001172)	123.96582260	0.00395446	0.9	90	25	30.7
12800(0.000391)	30720(0.000586)	123.96519554	0.00062707	6.3	166	32	40.4

In Tables 5.13 and 5.14, the transformation $S = 100e^x$ is used, and the x domain is $[-16, 2]$. The simple Convertible Bond shows relatively stable convergence speed with average ratio of 2.05 and a standard deviation of 0.08; but the full-featured Convertible Bond reveals unstable convergence, and the ratio is from 0.9 to 6.3. The average ratio is 2.7, and the standard deviation is 1.9. The order of convergence often oscillates.

The average number of iterations to converge required by the American option is approximately 11.5, while it is 22.5 for the simple Convertible Bond, and 40.4 for the full-featured Convertible Bond. This indicates that the Convertible Bond requires more iterations.

5.3.2 Penalty Method Results

The PSOR method is efficient for the problems with few discontinuities; however, when used in complex problems, it becomes slow and inefficient. It calculates the continuous value first, and applies the constraints to the derivative result later. It consumes many iterations, and the number of iterations increases fast when the timestep decreases.

The penalty method applies the constraints to the PDE formula, and can be used for complicated problems which may have several discontinuities. It requires fewer iterations than the PSOR method, and the number of iterations is relatively stable as the timestep is reduced. Compared to the PSOR method, it saves considerable number of iterations, and therefore it solves the problem more efficiently.

American Option

Table 5.15 lists the numerical results for an American put option using the penalty method associated with the Crank-Nicolson method. The convergence ratio is 3.9, which is consistent with a second order method. Notice that only few timesteps use more than one iteration.

Table 5.15: Results of penalty method for an American put option

TimeSteps($\Delta\tau$)	Nodes(ΔS)	Price	Difference	Ratio	Iters	Time
100(2.500000e-03)	26(2.000000e+01)	14.16785103			103	0s
200(1.250000e-03)	51(1.000000e+01)	14.54745964	0.37960861		205	0s
400(6.250000e-04)	101(5.000000e+00)	14.64501155	0.09755191	3.9	409	0s
800(3.125000e-04)	201(2.500000e+00)	14.67024098	0.02522942	3.9	818	1s
1600(1.562500e-04)	401(1.250000e+00)	14.67668852	0.00644754	3.9	1637	3s
3200(7.812500e-05)	801(6.250000e-01)	14.67832417	0.00163565	3.9	3274	12s
6400(3.906250e-05)	1601(3.125000e-01)	14.67873773	0.00041357	4.0	6549	45s
12800(1.953125e-05)	3201(1.562500e-01)	14.67884244	0.00010471	3.9	13105	209s

Table 5.15 uses the same parameters and relationships between $\Delta\tau$ and ΔS as Table 5.12. For comparison purposes, Table 5.16 lists the total number of iterations required

by the PSOR and penalty methods for the American put option.

Table 5.16: Comparison of PSOR and penalty method (American put option)

	PSOR Method		Penalty Method		Percentage
TimeSteps($\Delta\tau$)	Price	Iters	Price	Iters	Saved
100(2.500000e-03)	14.16785097	510	14.16785103	103	80%
200(1.250000e-03)	14.54745925	1060	14.54745964	205	81%
400(6.250000e-04)	14.64501132	2200	14.64501155	409	81%
800(3.125000e-04)	14.67024054	4880	14.67024098	818	83%
1600(1.562500e-04)	14.67668766	11040	14.67668852	1637	85%
3200(7.812500e-05)	14.67832267	24960	14.67832417	3274	87%
6400(3.906250e-05)	14.67873688	56960	14.67873773	6549	89%
12800(1.953125e-05)	14.67883348	147200	14.67884244	13105	91%

Table 5.16 shows that as the timestep and mesh size are reduced, the saved percentage ($\frac{\text{PSOR Iters - Penalty Iters}}{\text{PSOR Iters}} \times 100\%$) of number of iterations increases. For example, at the timestep level of 1.95×10^{-5}, as the approximate price is approaching the correct result, the PSOR method uses 147200 iterations; while the penalty method uses only 13105 iterations. Therefore, 91% of iterations are saved, making the penalty method substantially more efficient than PSOR.

Rannacher smoothing [6] is regarded as a second order convergence guarantee for the Crank-Nicolson method to deal with parabolic PDEs with nonsmooth initial conditions. In the Rannacher scheme, we take a few fully implicit steps (refereed to as *Rannacher steps* or *Rannacher smoothings*) after each nonsmooth initial state, and then use Crank-Nicolson thereafter. It should be noted that, if the number of fully implicit steps is small and independent of the total number of timesteps, the second order convergence arising from the Crank-Nicolson method is preserved. Even though second order convergence does not guarantee that the solution is non-oscillatory, this method works well in practice.

In order to test how many implicit steps are needed to smooth down the discontinuity, one, two and four implicit step(s) are used and the results are shown in Table 5.17.

In Table 5.17, the relationship between $\Delta\tau$ and ΔS is $\Delta\tau = 1.25 \times 10^{-4}\Delta S$, and "R. Smoothings" denotes the number of steps that use Rannacher smoothing. The convergence ratio is 3.9, which corresponds to a little bit less than quadratic order. It is observed that the ratios, the number of iterations, and also the running time are similar for each of the different numbers of Rannacher steps. Taking into account that the correct answer is close to 14.6788 [6], one Rannacher step (one step of the fully implicit method) is enough; more than one steps do not help substantially. Moreover, comparing the results of Tables 5.15 and 5.17, we see that the benefit from having one Rannacher step compared to having no Rannacher steps at all is minimal: only a few iterations are saved. This is because the American put option price does not have many discontinuities. However, the Convertible Bond price has several discontinuities, and therefore, as will be argued further in the thesis, Rannacher smoothing for Convertible Bond pricing is important. In the later numerical results on American options, one Rannacher step is applied.

Variable timestep

In all results so far, we use a constant timestep. The timestep is

$$\Delta\tau = \frac{T - t}{N}, \tag{5.4}$$

where T is the maturity time, t is the start time or current time, and N is the number of timesteps. While we increase N, we are reducing $\Delta\tau$.

Constant timestep treats every time interval the same. In the real problem, this is not true, because more complicated problems usually have more discontinuities. In the smooth areas, having a fine timestep does not help. A time selector is used to adjust the timestep [6].

Table 5.17: Results with different number of Rannacher steps used with the penalty method for an American put option

R. Smoothings	TimeSteps	Nodes	Price	Difference	Ratio	Iters	Time
1	100	26	14.16709136			102	0s
	200	51	14.54680556	0.37971420		203	0s
	400	101	14.64488489	0.09807932	3.9	408	0s
	800	201	14.67016779	0.02528291	3.9	816	1s
	1600	401	14.67665710	0.00648930	3.9	1635	3s
	3200	801	14.67831088	0.00165378	3.9	3271	11s
	6400	1601	14.67873258	0.00042170	3.9	6546	43s
	12800	3201	14.67884034	0.00010776	3.9	13099	174s
2	100	26	14.16647146			102	0s
	200	51	14.54668061	0.38020915		204	0s
	400	101	14.64478600	0.09810539	3.9	408	0s
	800	201	14.67013048	0.02534447	3.9	816	1s
	1600	401	14.67663930	0.00650883	3.9	1634	2s
	3200	801	14.67830357	0.00166427	3.9	3271	10s
	6400	1601	14.67872924	0.00042567	3.9	6546	45s
	12800	3201	14.67883904	0.00010980	3.9	13099	182
4	100	26	14.16515905			102	0s
	200	51	14.54640253	0.38124349		204	0s
	400	101	14.64462995	0.09822742	3.9	408	0s
	800	201	14.67006429	0.02543434	3.9	816	1s
	1600	401	14.67661154	0.00654725	3.9	1634	3s
	3200	801	14.67829206	0.00168051	3.9	3272	11s
	6400	1601	14.67872444	0.00043239	3.9	6547	41s
	12800	3201	14.67883712	0.00011267	3.8	13100	178s

Given an initial timestep $\Delta\tau^{n+1}$, a new timestep is selected so that

$$\Delta\tau^{n+2} = \left(\min_i \left[\frac{dnorm}{\frac{|V(S_i,\tau^n+\Delta\tau^{n+1})-V(S_i,\tau^n)|}{\max(D,|V(S_i,\tau^n+\Delta\tau^{n+1})|,|V(S_i,\tau^n)|)}} \right] \right) \Delta\tau^{n+1}, \tag{5.5}$$

where $dnorm$ is a target relative change (during the timestep) specified by the user. The denominator $\frac{|V(S_i,\tau^n+\Delta\tau^{n+1})-V(S_i,\tau^n)|}{\max(D,|V(S_i,\tau^n+\Delta\tau^{n+1})|,|V(S_i,\tau^n)|)}$ in the above formula has been normalized by $\max(D, |V(S_i, \tau^n + \Delta\tau^{n+1})|, |V(S_i, \tau^n)|)$ for two reasons. First, the scale D is selected so that the timestep selector does not take an excessive number of timesteps in regions where the value is small. For options valued in dollars, $D = 1$ is typically appropriate [6]. Second, normalization by $|V(S_i, \tau^n + \Delta\tau^{n+1})|$ or $|V(S_i, \tau^n)|$ avoids slow timestep growth for large values of the contract.

The timestep selector (5.5) estimates the change in the solution at the new timestep based on changes observed over the old timestep. We choose a $(\Delta\tau)^0$ for the coarsest grid, and then $(\Delta\tau)^0$ is divided by four at each grid refinement. If the timestep is too conservative, the timestep will increase rapidly; so there would be no problem if $(\Delta\tau)^0$ is underestimated. In the American option examples, we chose $(\Delta\tau)^0 = 10^{-3}$ and $dnorm = 0.2$ on the coarsest grid. The value of $dnorm$ was reduced by 2 at each grid refinement.

Table 5.18 presents the results using the timestep selector, and Table 5.19 the results using constant timestep. In both tables, the spatial grid points are uniform.

Comparing these two tables, if we take the timestep level of 1.5625×10^{-5}, we see that with the same number of spatial grid points, Table 5.18, which uses the timestep selector, takes 531 iterations in total; while Table 5.19, which uses the constant timestep, takes 16190 iterations, which is about 30 times more. The former obtains the result in 27 seconds, and the latter in 604 seconds. The former saves about 95.5% of the time. The convergence ratio is almost the same, 4.1. Therefore, we conclude that the timestep selector helps save time dramatically without impacting the convergence ratio.

Table 5.18: Results of penalty method for an American put option (timestep selector)

TimeSteps($\Delta\tau$)	Nodes(ΔS)	Price	Diff	Ratio	Iters	Time
39(1.000000e-03)	251(2.000000e+00)	14.67357879			61	0s
83(2.500000e-04)	501(1.000000e+00)	14.67761401	0.00403522		127	1s
170(6.250000e-05)	1001(5.000000e-01)	14.67856975	0.00095574	4.2	262	6s
342(1.562500e-05)	2001(2.500000e-01)	14.67880218	0.00023243	4.1	531	27s
686(3.906250e-06)	4001(1.250000e-01)	14.67885950	0.00005732	4.1	1067	137s
1372(9.765625e-07)	8001(6.250000e-02)	14.67887366	0.00001416	4.0	2137	749s

Table 5.19: Results of penalty method for an American put option (constant timestep)

TimeSteps($\Delta\tau$)	Nodes(ΔS)	Price	Diff	Ratio	Iters	Time
250(1.000000e-03)	251(2.000000e+00)	14.67278666			269	2s
1000(2.500000e-04)	501(1.000000e+00)	14.67738944	0.00460278		1042	9s
4000(6.250000e-05)	1001(5.000000e-01)	14.67851251	0.00112307	4.1	4091	77s
16000(1.562500e-05)	2001(2.500000e-01)	14.67878794	0.00027543	4.1	16190	604s

Convertible Bond

In order to carry out a careful convergence study of the Convertible Bond, we investigate the simple Convertible Bond with no other feature but only convertibility, and proceed gradually to the more complicated cases: coupon payment, puttability, and callability. The CB price is partially determined by the COCB component, and the COCB itself also follows the same process of CB. Thus, smoothing discontinuities becomes very important. In order to do that, four substeps are used for each timestep. The first substep uses the fully implicit scheme functioning as a Rannacher smoothing, and the other three use the Crank-Nicolson scheme. The fully implicit step is for smoothing the discontinuity for both the COCB and the CB. Table 5.20 has the results for the simple Convertible Bond and the four substeps technique.

Table 5.20: Results of penalty method for a simple CB

TimeSteps($\Delta\tau$)	Nodes(ΔS)	Price	Diff	Ratio	Iters
100(5.000000e-02)	101(5.000000e+00)	104.11408790			700
200(2.500000e-02)	201(2.500000e+00)	104.20544884	0.09136094		1400
400(1.250000e-02)	401(1.250000e+00)	104.24753429	0.04208545	2.2	2800
800(6.250000e-03)	801(6.250000e-01)	104.26758582	0.02005154	2.1	5600
1600(3.125000e-03)	1601(3.125000e-01)	104.27706862	0.00948280	2.1	11200
3200(1.562500e-03)	3201(1.562500e-01)	104.28178825	0.00471963	2.0	22400
6400(7.812500e-04)	6401(7.812500e-02)	104.28413712	0.00234887	2.0	44800
12800(3.906250e-04)	12801(3.906250e-02)	104.28530755	0.00117043	2.0	89600

From Table 5.20 we can see that even the simple Convertible Bond cannot achieve the convergence ratio of the American option. The average ratio is 2.1, and the standard deviation is 0.08. This is close to first order convergence with respect to $\Delta\tau$. The following factors may be the reason: the convertibility in the whole life of the bond brings complexity to the Convertible Bond; the method we are using is not the traditional second order method; a fully implicit (sub)step is used for smoothing at each timestep. The convergence ratio becomes even worse for the Convertible Bond with other features. This is reflected in Tables 5.21 to 5.23, which include the results for a more complicated Convertible Bond.

Table 5.21: Results of penalty method for a simple CB plus coupon payment

TimeSteps($\Delta\tau$)	Nodes(ΔS)	Price	Diff	Ratio	Iters
100(5.000000e-02)	101(5.000000e+00)	135.55245684			630
200(2.500000e-02)	201(2.500000e+00)	135.47984504	0.07261180		1260
400(1.250000e-02)	401(1.250000e+00)	135.43890856	0.04093648	1.8	2520
800(6.250000e-03)	801(6.250000e-01)	135.45852423	0.01961567	2.1	5040
1600(3.125000e-03)	1601(3.125000e-01)	135.46796906	0.00944483	2.1	10080
3200(1.562500e-03)	3201(1.562500e-01)	135.46272850	0.00524056	1.8	20160
6400(7.812500e-04)	6401(7.812500e-02)	135.46009306	0.00263544	2.0	41629
12800(3.906250e-04)	12801(3.906250e-02)	135.46126500	0.00117194	2.2	91929

Table 5.21 shows the numerical results for the simple Convertible Bond with coupon payment; Table 5.22 shows the numerical results with an extra "put" feature; and Table 5.23 shows the numerical results for the full-featured Convertible Bond, with an extra "call" feature compared to the previous one. For the purpose of comparison, we list the average ratio, the standard deviation, and the number of iterations at timestep level 3.906250e-04 in Table 5.24.

From Table 5.24, we observe that the simple Convertible Bond with or without coupon payment and/or puttability has first order convergence with respect to $\Delta\tau$; the full-featured Convertible Bond has less than first order. The order corresponding to the Convertible Bond with put and/or call features oscillates more than that without these

Table 5.22: Results of penalty method for a simple CB with coupon and puttability

TimeSteps($\Delta\tau$)	Nodes(ΔS)	Price	Diff	Ratio	Iters
100(5.000000e-02)	101(5.000000e+00)	135.72285238			520
200(2.500000e-02)	201(2.500000e+00)	135.64745544	0.07539693		1040
400(1.250000e-02)	401(1.250000e+00)	135.60427599	0.04317946	1.7	2080
800(6.250000e-03)	801(6.250000e-01)	135.61958572	0.01530973	2.8	4161
1600(3.125000e-03)	1601(3.125000e-01)	135.62672976	0.00714404	2.1	8321
3200(1.562500e-03)	3201(1.562500e-01)	135.62230840	0.00442136	1.6	16642
6400(7.812500e-04)	6401(7.812500e-02)	135.62006574	0.00224266	2.0	33284
12800(3.906250e-04)	12801(3.906250e-02)	135.62093688	0.00087114	2.6	66565

Table 5.23: Results of penalty method for a full-featured CB

TimeSteps($\Delta\tau$)	Nodes(ΔS)	Price	Diff	Ratio	Iters
100(5.000000e-02)	101(5.000000e+00)	124.10689260			752
200(2.500000e-02)	201(2.500000e+00)	124.04697190	0.05992070		1577
400(1.250000e-02)	401(1.250000e+00)	124.01543469	0.03153721	1.9	3347
800(6.250000e-03)	801(6.250000e-01)	123.99811624	0.01731845	1.8	6500
1600(3.125000e-03)	1601(3.125000e-01)	123.98745588	0.01066036	1.6	12577
3200(1.562500e-03)	3201(1.562500e-01)	123.98002499	0.00743089	1.4	24974
6400(7.812500e-04)	6401(7.812500e-02)	123.97501252	0.00501248	1.5	49975
12800(3.906250e-04)	12801(3.906250e-02)	123.97143882	0.00357370	1.4	99913

Table 5.24: Comparison of penalty method performance for different-featured CBs

Feature(s)	Average Ratio	Standard Deviation	No. of Iterations
Simple	2.1	0.08	89600
Simple + Coupon	2.0	0.17	91929
Simple + Coupon +Put	2.1	0.48	66565
Simple + Coupon +Put +Call	1.6	0.21	99913

features. The full-featured Convertible Bond consumes more iterations than the others.

In order to compare the two iterative methods, we list the total number of iterations required for the full-featured Convertible Bond by both the PSOR and penalty methods in Table 5.25.

Table 5.25: Comparison of PSOR and penalty method (full-featured CB)

	PSOR Method		Penalty Method		Percentage
TimeSteps($\Delta\tau$)	Price	Iters	Price	Iters	Saved
200(2.500000e-02)	124.07379824	1800	124.04697190	1577	12.4%
400(1.250000e-02)	124.00140099	4440	124.01543469	3347	24.6%
800(6.250000e-03)	123.98866460	10880	123.99811624	6500	40.3%
1600(3.125000e-03)	123.97740160	27840	123.98745588	12577	54.8%
3200(1.562500e-03)	123.96923912	72640	123.98002499	24974	65.6%
6400(7.812500e-04)	123.96650281	191360	123.97501252	49975	73.9%
12800(3.906250e-04)	123.96529799	504320	123.97143882	99913	80.2%

Table 5.25 reveals an increasing trend of percentage in number of iterations saved, similar to that in Table 5.16 for an American option. While for coarse grids and large timesteps the percentage saved is not as significant as the respective one for the American

options, as the grid is refined and the timestep becomes smaller, the percentage saved is almost as significant as the respective one for the American options. Compared to the PSOR method, the penalty iteration is a better method for both the American option and the Convertible Bond pricing problems in terms of the number of iterations.

In order to show the importance of Rannacher smoothing for Convertible Bond pricing, we present, in Table 5.26, the results of pricing the full-featured Convertible Bond with the same parameters and the same four-substep procedure as the results of Table 5.23, except that all four substeps are Crank-Nicolson steps. We notice that these results

Table 5.26: Results of penalty method for a full-featured CB without Rannacher smoothing

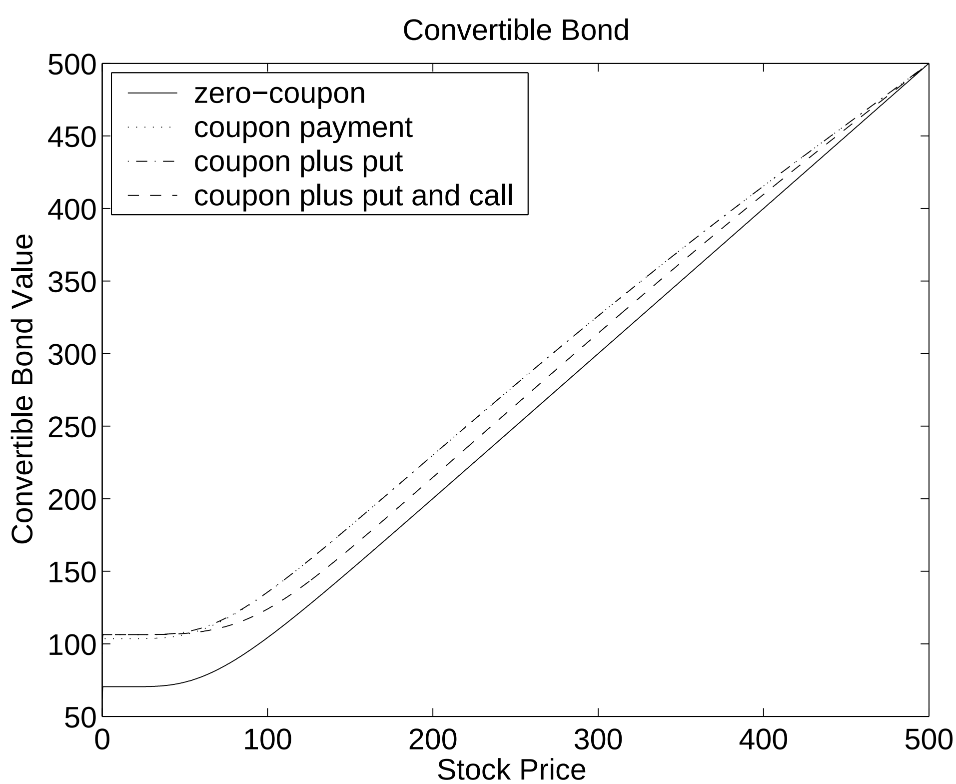
TimeSteps($\Delta\tau$)	Nodes(ΔS)	Price	Diff	Ratio	Iters
100(5.000000e-02)	101(5.000000e+00)	123.71747423			809
200(2.500000e-02)	201(2.500000e+00)	123.27755037	0.43992386		1610
400(1.250000e-02)	401(1.250000e+00)	123.14507781	0.13247256	3.3	3215
800(6.250000e-03)	801(6.250000e-01)	122.77961626	0.36546154	0.4	6427
1600(3.125000e-03)	1601(3.125000e-01)	123.03085648	0.25124021	1.5	12835
3200(1.562500e-03)	3201(1.562500e-01)	123.00333432	0.02752216	9.1	25656
6400(7.812500e-04)	6401(7.812500e-02)	122.98866071	0.01467361	1.9	51302

oscillate noticeably. While, in Table 5.23, the “correct” result (123.97) is obtained with 6400 timesteps, with the same number of timesteps in Table 5.26 convergence has not been obtained, and the Convertible Bond price does not clearly converge to the correct result. Thus, Rannacher smoothing is important in Convertible Bond pricing.

5.4 Plots

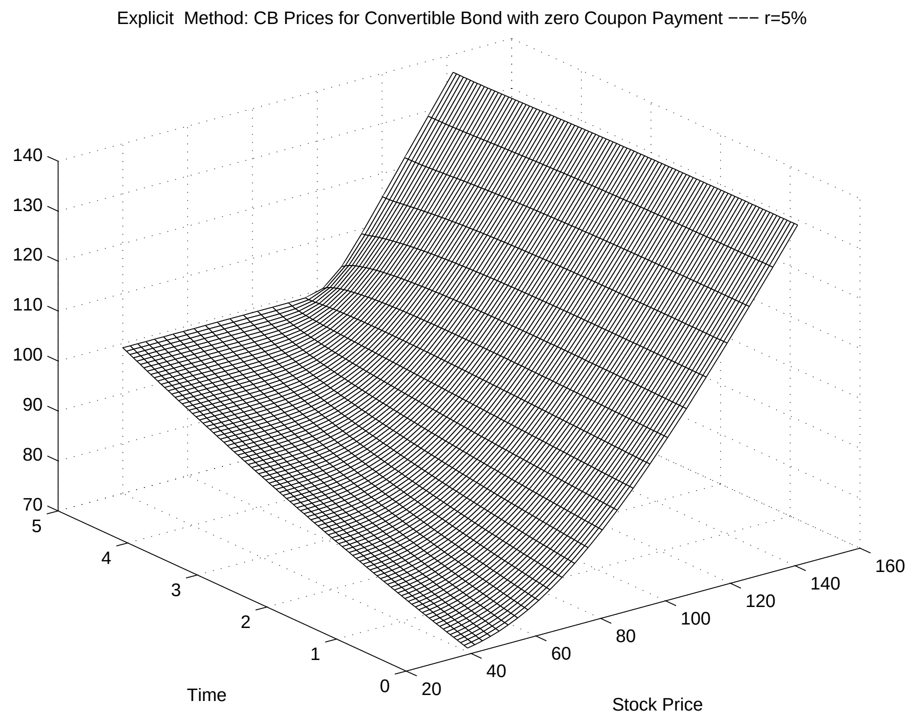
Convertible bonds with different features have different prices. Figure 5.1 exhibits the difference. The simple Convertible Bond has the lowest price among the four types of Convertible Bonds plotted. Coupon payments increase the value of the Convertible Bond. The put feature is in favor of the holder, and thus also increases the value of the Convertible Bond. The call feature is against the holder and in favor of the issuer, and it thereby reduces the value. It is also worth mentioning that, because of the convertibility, the simple Convertible Bond has a lower price than the straight bond.

Figure 5.1: Price comparison for Convertible Bonds with different features

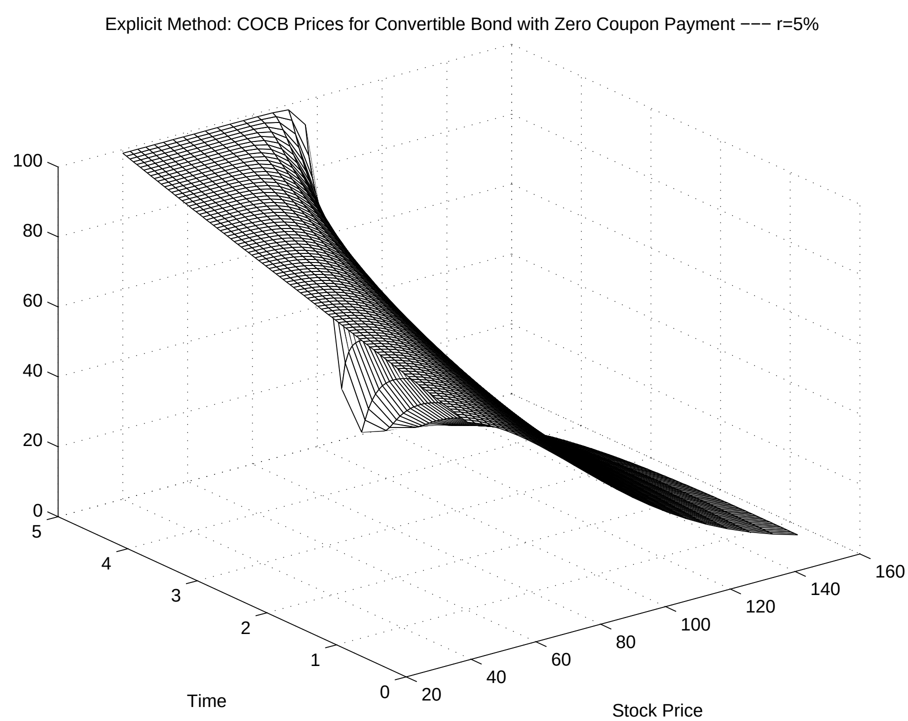


In Figures 5.2 to 5.7, we present three-dimensional plots of the Convertible Bond prices with different features versus the stock price and the time. For simplicity, we use the results from the explicit method.

Figure 5.2: Plots for zero-coupon bond (the simple Convertible Bond) with $r = 5\%$

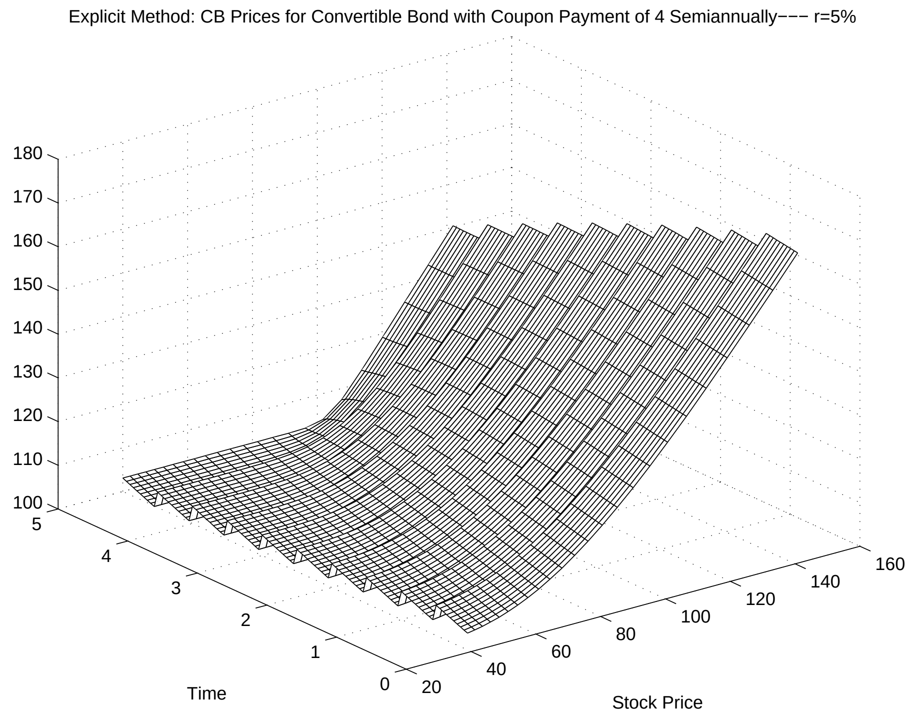


(a) Zero CB

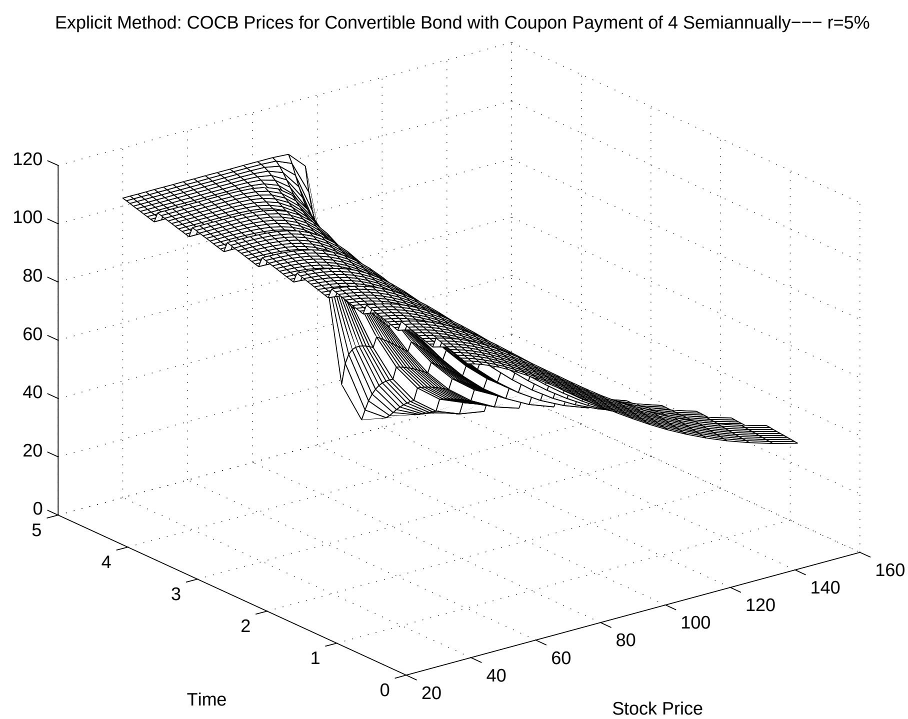


(b) Zero COCB

Figure 5.3: Plots for a coupon payment only Convertible Bond with $r = 5\%$

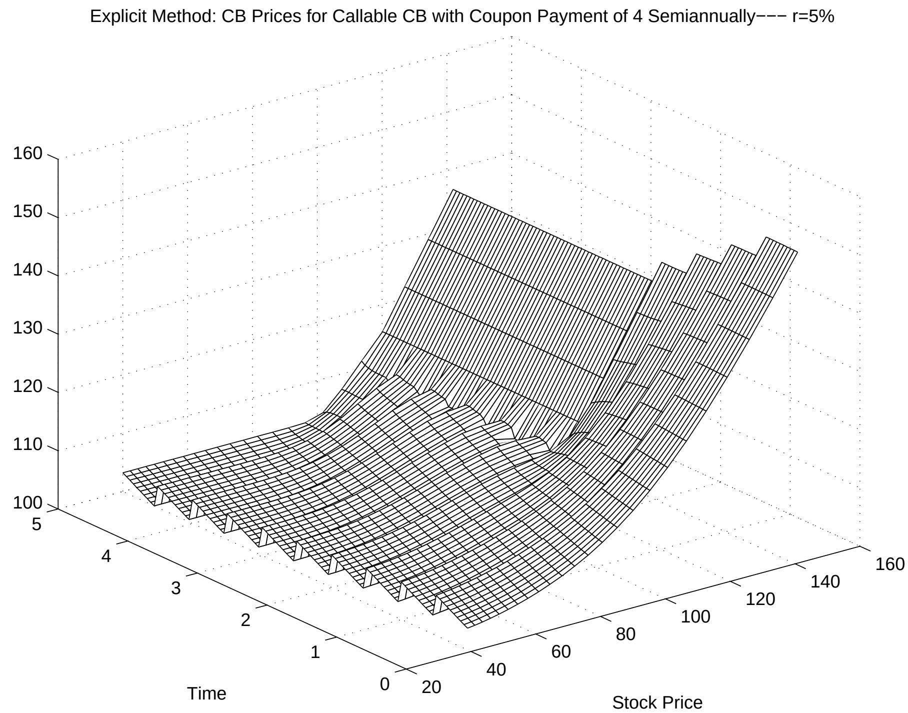


(a) Coupon CB

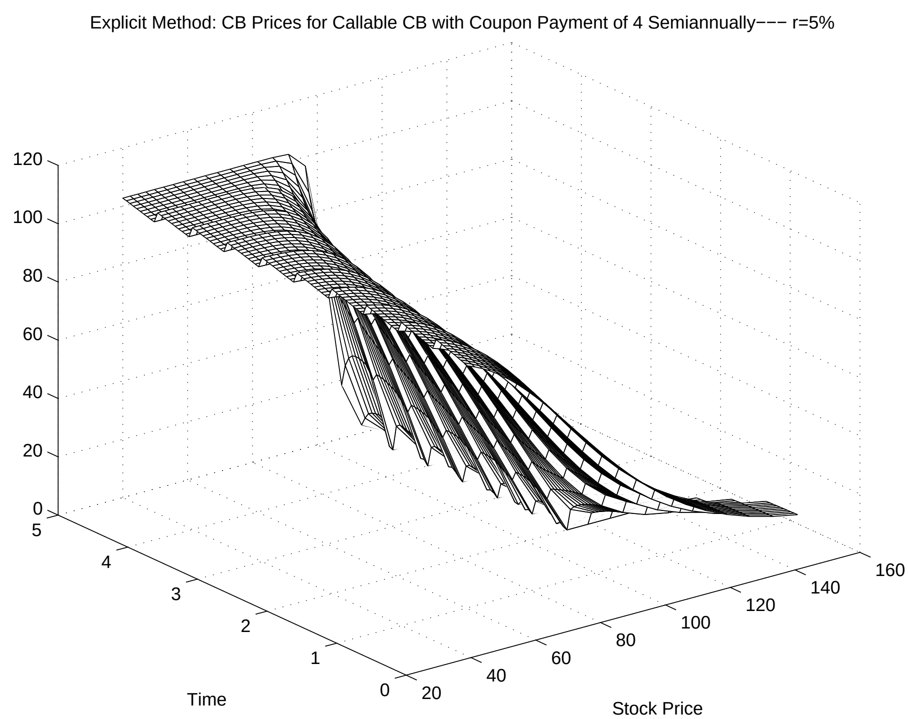


(b) Coupon COCB

Figure 5.4: Plots for a coupon payment and callable Convertible Bond with $r = 5\%$

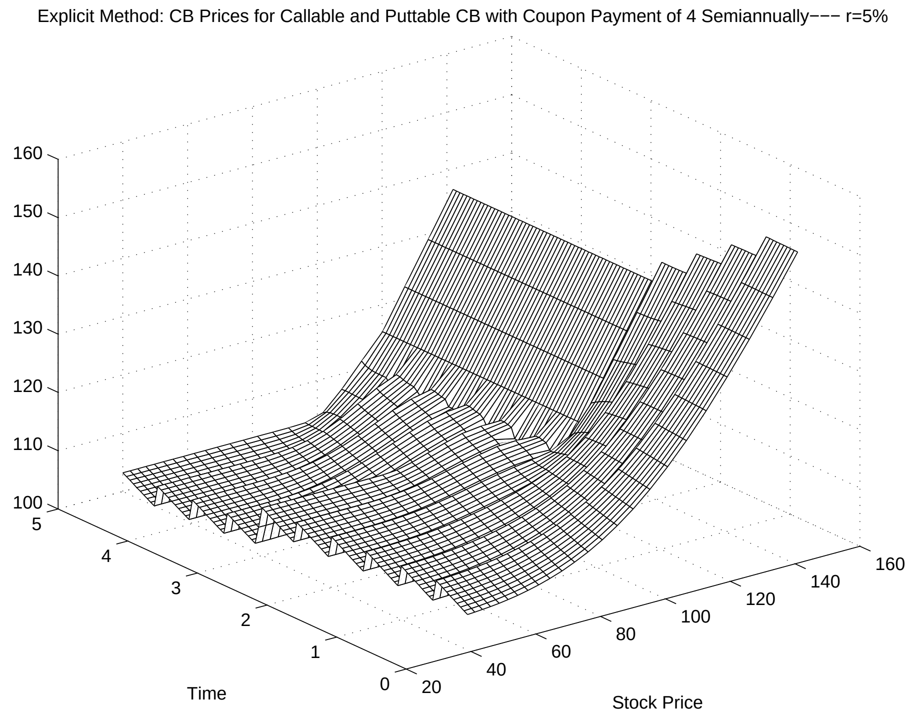


(a) Coupon + call CB

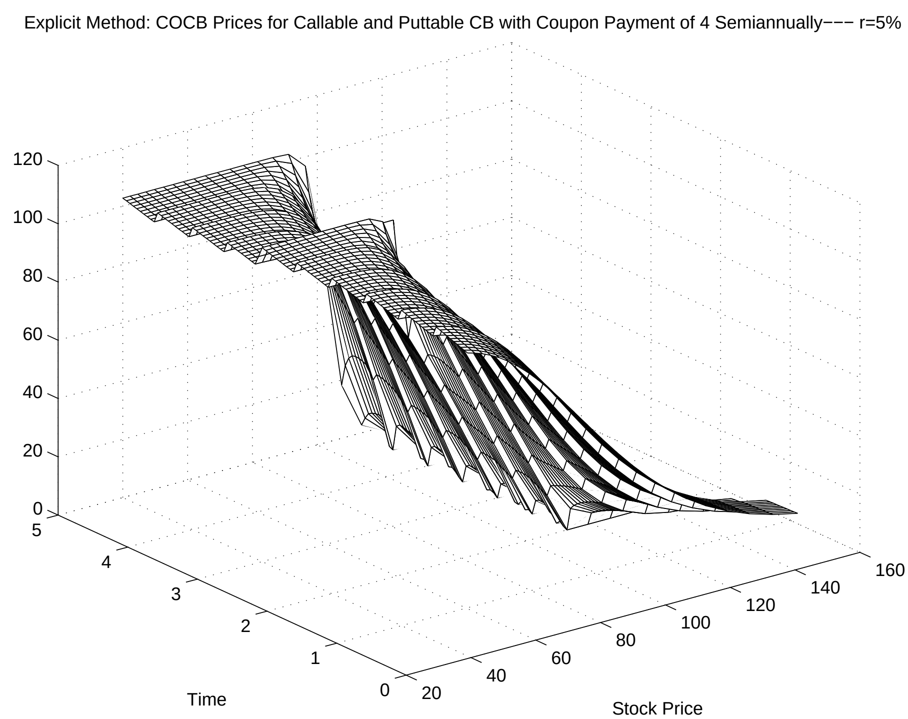


(b) Coupon + call COCB

Figure 5.5: Plots for a full-featured Convertible Bond with $r = 5\%$



(a) Full-featured CB



(b) Full-featured COCB

The TF paper [16], unfortunately, does not give quantitative results. Hence, we only check the similarities and differences between these plots and those in the TF paper. The TF paper also has eight plots showing similar quantities.

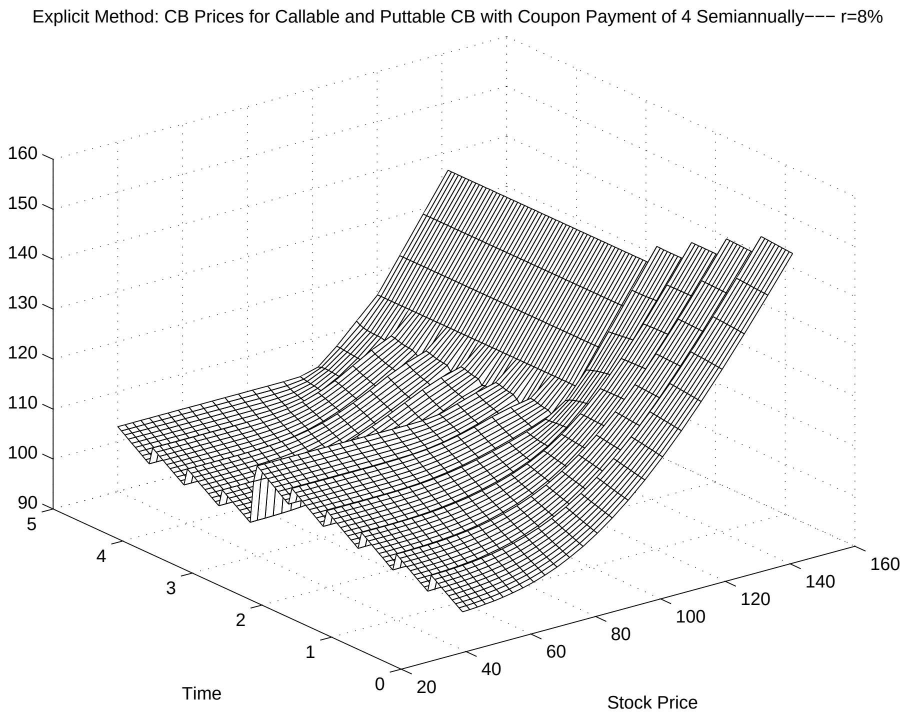
Regarding the plots in Figures 5.2 to 5.5, those without the call feature are consistent with the respective ones in the TF paper; however, those plots involving the call feature are different from TF's. Notice that the plots in Figures 5.4 and 5.5 have no ripples in the region with high equity level after a certain time, while TF's plots have ripples all over. See, for example, the plot in page 5 and the top-left plot in page 7 in the TF paper.

The reason the plots in Figures 5.4 and 5.5 have no ripples in the region with high equity level is that the Convertible Bond has no optionality in that region, as conversion will have occurred. Hence the price of the Convertible Bond is proportional to that of the stock price, and has nothing to do with the bond.

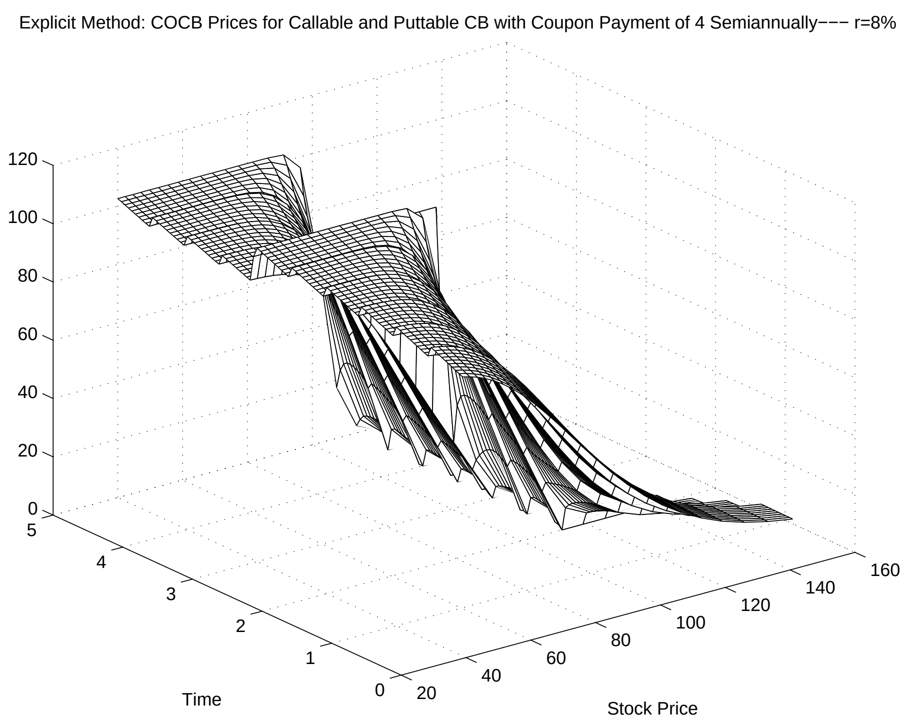
In Figures 5.6 and 5.7, we present three-dimensional plots of the full-featured Convertible Bond price versus the stock price and the time, with interest rates 8% and 10%, respectively. Note that Figure 5.5 plots of the full-featured Convertible Bond price versus the stock price and the time, with interest rates 5%. Figures 5.5, 5.6 and 5.7 show jumps at the end of the put exercise time (end of third year) and at the low asset price range. This is expected, because (a) the put feature increases the value of the bond, but after the end of the put exercise time, there is no more chance to exercise the put, therefore this extra value disappears, and (b) the put feature is exercised only when the value of the bond falls below B_p. Notice that the third year is between $t = 2$ and $t = 3$.

Notice also that the jumps increase as the interest rate is increasing. The reason is that, when the interest rate rises, bond prices tend to fall, and therefore, after the put option is realized (and the extra value disappears), the bond price with a high r falls at a lower level than the bond price with a low r.

Figure 5.6: Plots for a full-featured Convertible Bond with $r = 8\%$

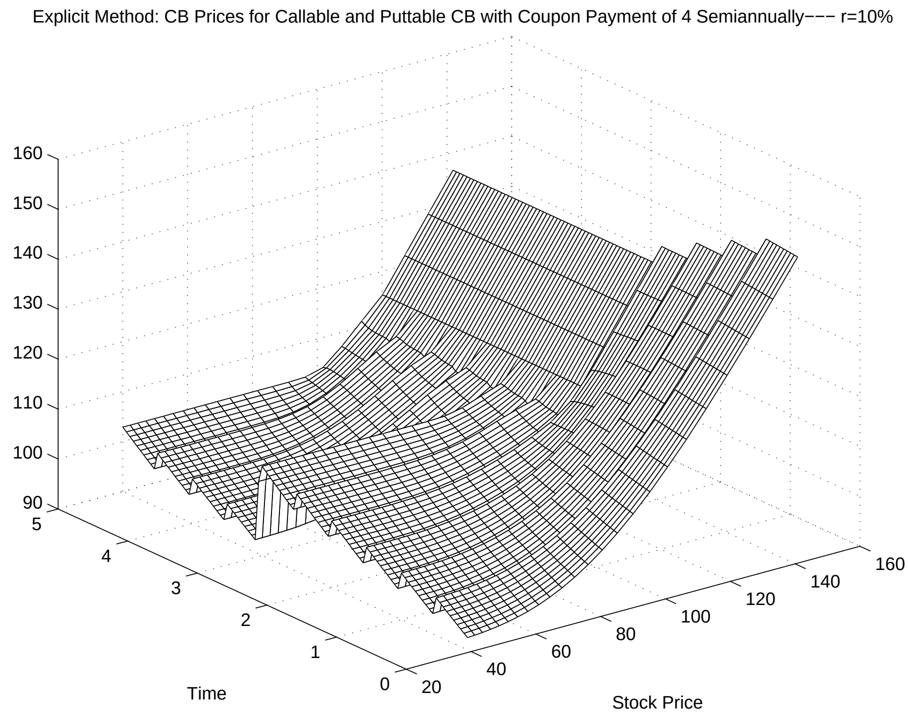


(a) Full-featured CB

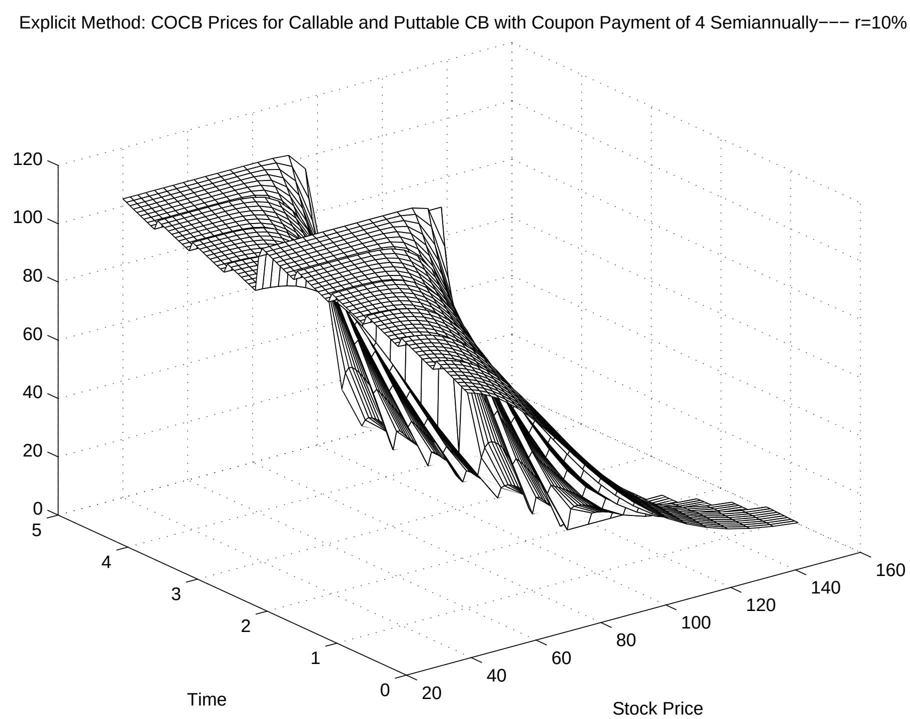


(b) Full-featured COCB

Figure 5.7: Plots for a full-featured Convertible Bond with $r = 10\%$



(a) Full-featured CB



(b) Full-featured COCB

Chapter 6

Conclusions and Future Work

The Convertible Bond is a hybrid security. It can be converted into equity; it may be purchased back by the issuer in the future at a specified price; and it is also possible to be sold back by the holder to the issuer in a certain time at a pre-determined price. The call and put features increase the complexity of pricing the Convertible Bond.

The TF model decomposes the Convertible Bond into two parts: the cash-only part — COCB, and the risk-free equity part. It couples the pricing of the COCB into the pricing of the entire Convertible Bond. We implemented this model using the explicit method and several variations of the implicit method coupled with two iterative approaches.

In order to study the convergence property, we always started from simple cases. We studied a method on an American put option first, then applied it to the simple Convertible Bond, and finally put it into the full-featured Convertible Bond that has coupon payment, convertibility, puttability and callability.

The explicit finite difference method, which is a first order method with respect to $\Delta\tau$, requires strict relationships between $\Delta\tau$ and ΔS or Δx for both stability and convergence. For the American option, the experimentally observed convergence order of the explicit method is consistent with the theoretical one. However, for the Convertible Bond, the explicit method is inefficient and exhibits oscillatory convergence order. The convergence order of the Convertible Bond is lower than that of the American option because the former is much more complicated than the latter.

We explored both the PSOR and penalty iteration associated with the Crank-Nicolson and the fully implicit methods. The PSOR method achieved the second order convergence for the American option, but it consumed many iterations. The PSOR method is not efficient for complicated derivatives, e.g. the Convertible Bond. In order to reduce computational costs, we studied the penalty method which can considerably reduce the number of iterations. For the American option, most of the timesteps used one iteration, and only few steps used more than one but at most two iterations. For the Convertible Bond, almost all the timesteps used more than two iterations, but, still, the penalty method saved many iterations and much computation time compared to the PSOR method. As the timestep and the grid interval reduced, the percentage of the saved iterations increased.

We also studied the effect of using variable timesteps. An adaptive timestep selector technique helped reduce the computational cost dramatically, while the convergence ratio did not degrade.

We plotted the Convertible Bond price. We found that our plots with call features were different from those in the TF paper. Our plots have no ripples in the region at high equity level, while those in the TF paper have ripples all over. We argue that at high equity levels when the Convertible Bond trades at parity, there is no optionality as conversion will have occurred, and therefore, there would be no ripples in this area.

We believe the method for pricing the Convertible Bond can be improved in the following aspects. First, we may use different interpolation methods to improve the accuracy of the results. Second, we can consider using another explicit scheme, the second order scheme, to improve the convergence speed. We may also consider a coupled Linear Complementarity Problem formula, in which both COCB and CB are coupled together in one big matrix, and therefore can be solved together in one iteration.

Apart from the features we considered in this thesis, some Convertible Bonds have a "dividend protection" feature. Since a big dividend will cause a share price decrease, when the company increases its dividend on the common stock, the bond should become convertible to more shares. In the future, we may consider including the dividend rate protection to the Convertible Bond pricing.

Bibliography

[1] E. Ayache, P. A. Forsyth, and K. R. Vetzal. The valuation of convertible bonds with credit risk. *Journal of Derivatives*, fall(11):9–29, 2003.

[2] Martin Baxter. *Financial calculus: an introduction on derivative pricing.* Cambridge University Press, 1996.

[3] M. J. Brennan and E. S. Schwartz. Convertible bonds: Valuation and optimal strategies for call and conversion. *The Journal of Finance*, 32:1699–1715, 1997.

[4] Michael J. Brennan and Eduardo S. Schwartz. Analyzing convertible bonds. *Journal of Financial and Quantitative Analysis*, 15:907–929, 1980.

[5] Y. D'Halluin, P. A. Forsyth, K. R. Vetzal, and G. Labahn. A numerical PDE approach for pricing callable bonds. *Applied Mathematical Finance*, 8:49–77, 2001.

[6] P. A. Forsyth and K. R. Vetzal. Quadratic convergence for valuing American options using a penalty method. *SIAM J. Sci. Comput.*, 23(6):2095–2122, 2002.

[7] Lawrence Galitz. *Financial engineering: tools and techniques to manage financial risk.* IRWIN Professional Publishing, 1995.

[8] Russell Grimwood and Stewart Hodges. The valuation of convertible bonds: A study of alternative pricing models. *Warwick Business School Pre-print PP02-121,* `http://www2.warwick.ac.uk/`, 2002.

[9] Thomas S. Y. Ho and David M. Pfeffer. Convertible bonds: Model, value attribution, and analytics. *Financial Analysts Journal*, Sep/Oct(52):35–44, 1996.

[10] John Hull. *Options, Futures, and Other Derivatives (5th edition).* Prentice Hall, 2002.

[11] Peter E. Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations.* Springer, 1992.

[12] A. R. Mitchell and D. F. Griffiths. *The Finite Difference Method in Partial Differential Equations.* John Wiley & Sons, 1987.

[13] K. G. Nyborg. The use and pricing of convertible bonds. *Applied Mathematical Finance*, 3:167–190, 1996.

[14] D. M. Pooley, P. A. Forsyth, and K. R. Vetzal. Numerical convergence properties of option pricing PDEs with uncertain volatility. *IMA Journal of Numerical Analysis*, 23:241–267, 2003.

[15] Domingo Tavella and Curt Randall. *Pricing Financial Instruments: the finite Difference Method.* John Wiley& Sons, Inc., 2000.

[16] Kostas Tsiveriotis and Chris Fernandes. Valuing convertible bonds with credit risk. *The Journal of Fixed Income*, 8(2):95–102, 1998.

[17] Paul Wilmott. *Derivatives, the theory and practice of financial engineering.* John Wiley & Sons, Inc., 1998.

[18] Paul Wilmott, Sam Howison, and Jeff Dewynne. *The Mathematics of Financial Derivatives, A Student Introduction.* Cambridge University Press, 1995.

[19] Ali Bora Yigitbasioglu. Pricing convertible bonds with interest rate, equity, credit and FX risk. *ISMA Center, the University of Reading, Discussion paper 2001-14,* `http://www.ismacentre.rdg.ac.uk`, 2001.