HEC MONTRÉAL
École affiliée à l'Université de Montréal

Dynamic programming and parallel computing for valuing two-dimensional financial derivatives

par
Malek Ben Abdellatif

Thèse présentée en vue de l'obtention du grade de Ph. D. en Administration
(option Méthodes quantitatives)

Décembre 2016

HEC MONTRÉAL
École affiliée à l'Université de Montréal

Cette thèse intitulée :

Dynamic programming and parallel computing for valuing two-dimensional financial derivatives

présentée par :

Malek Ben Abdellatif

a été évaluée par un jury composé des personnes suivantes :

Georges Zaccour
HEC Montréal
Président-rapporteur

Hatem Ben Ameur
HEC Montréal
Codirecteur de recherche

Bruno Rémillard
HEC Montréal
Codirecteur de recherche

Nicolas Papageorgiou
HEC Montréal
Membre du jury

Mark Reesor
Wilfrid Laurier University
Examinateur externe

Georges Dionne
HEC Montréal
Représentant du directeur de HEC Montréal

RÉSUMÉ

Cette thèse présente trois essais portant sur l'évaluation des produits financiers à deux dimensions. Notre méthodologie est basée sur un programme dynamique à deux dimensions couplé aux éléments finis. Augmenter la dimension du problème rend cette procédure coûteuse en termes de temps d'exécution. Pour cette raison, nous utilisons le calcul parallèle pour améliorer l'efficacité.

Dans le premier essai, nous proposons un modèle d'évaluation pour les options américaines à deux dimensions. Notre modèle est flexible puisqu'il permet de considérer une large famille d'options écrites sur deux actifs qui suivent un processus lognormal. Nos investigations numériques montrent la convergence et l'efficience de notre méthodologie, ce qui la rend une alternative viable aux méthodologies traditionnelles.

Dans le second essai, nous développons un modèle structurel général à deux facteurs pour évaluer les dettes corporatives. La valeur des actifs de la firme suit un processus lognormal et le taux d'intérêt selon processus gaussien. Notre modèle permet une structure de dette flexible, plusieurs classes de séniorité, les économies de taxes ainsi que les coûts de faillite. Les résultats obtenus sont cohérents avec les effets empiriques documentés dans la littérature.

Dans le troisième essai, nous proposons un modèle structurel pour l'évaluation des obligations échangeables. Nous développons un modèle à deux dimensions où la valeur des actifs de la compagnie émettrice et la valeur des actions sous-jacentes représentent les deux variables d'état. Notre modèle permet une structure de dette arbitraire, les économies de taxes et les coûts de faillite. Notre investigation numérique souligne les principales caractéristiques des obligations échangeables.

Mots clés : Espace état à deux dimensions ; Programme dynamique ; Éléments finis ; Calcul Parallèle ; Options Américaines ; Risque de crédit ; Modèle structurel ; Taux d'intérêt stochastique ; Dettes échangeables.

Méthodes de recherche : Modélisation mathématique ; Recherche quantitative ; Calcul bivarié ; Méthodes numériques.

ABSTRACT

This thesis presents three essays on valuing two-dimensional financial securities. Our methodology is based on a two-dimensional dynamic program coupled with finite elements. As we have two state variables, this procedure is time consuming. Thus, we use parallel computing to enhance the efficiency.

In the first essay, we propose a valuation model for two-dimensional American-style options. Our model is flexible because it accommodates a large family of option contracts signed on two underlying assets that move according to a lognormal vector process. Our numerical experiments show convergence and efficiency which positions our method as a viable alternative to traditional methodologies.

In the second essay, we develop a general structural model for valuing risky corporate debts that takes into account both default and interest rate risk. We propose a two-dimensional model in which the state variables are the value of the firm's assets and the short-term interest rate. The former follows a lognormal process and the latter a mean-reverting Gaussian process. Our model accommodates flexible debt structure, multiple seniority classes, tax benefits, and bankruptcy costs. The results we obtain are consistent with empirical evidence documented in the literature.

In the third essay, we propose a model to value exchangeable bonds, which is a debt that is convertible into shares of a firm's equity other than the bond's issuer. We propose a two-dimensional structural model, where the assets' value of the bond's issuer and the underlying equity value are the state variables. Our model accommodates arbitrary debt portfolio, bankruptcy costs, and tax benefits. Our numerical investigation highlights the main characteristics of exchangeable bonds.

Keywords: Two-dimensional state spaces; Dynamic programming; Finite elements; Parallel computing; American options; Credit risk; Structural model; Stochastic interest rate; Exchangeable bond.

Research methods: Mathematical modeling; Quantitative research; Bivariate calculation; Numerical methods.

CONTENTS

RÉSUMÉ . . . . **iv**

ABSTRACT . . . . **v**

CONTENTS . . . . **vi**

LIST OF PAPERS . . . . **ix**

LIST OF TABLES . . . . **x**

LIST OF FIGURES . . . . **xi**

LIST OF APPENDICES . . . . **xiii**

DEDICATION . . . . **xiv**

ACKNOWLEDGMENTS . . . . **xv**

INTRODUCTION **1**

CHAPTER 1: DYNAMIC PROGRAMMING AND PARALLEL COMPUTING FOR VALUING TWO-DIMENSIONAL AMERICAN-STYLE OPTIONS **3**

Abstract **3**

1.1 Introduction 3

1.2 The model 5

1.3 Dynamic programming and parallel computing 7

1.3.1 Dynamic programming 7

1.3.2 Parallel computing 10

1.4 Numerical investigation 14

1.4.1 European options 14

1.4.2 American options 16
1.5 Conclusion 20

Appendix **22**
1.A Closed-form solutions for European options 22
1.A.1 Exchange option 22
1.A.2 Call-on-max option 22
1.A.3 Put-on-min option 23
1.B Transition parameters 24

Bibliography **27**

CHAPTER 2: A TWO-FACTOR STRUCTURAL MODEL FOR VALUING CORPORATE SECURITIES **30**

Abstract **30**
2.1 Introduction 30
2.2 Model and notations 33
2.3 Dynamic programming 38
2.4 Numerical investigation 39
2.5 Conclusion 44

Appendix **46**
2.A Forward measure 46
2.B Transitions parameters 48
2.C Parallel computing algorithm 50

Bibliography **52**

CHAPTER 3: A STRUCTURAL MODEL FOR VALUING EXCHANGEABLE BONDS **56**

Abstract **56**
3.1 Introduction 56
3.2 Valuation framework 59
3.3 Dynamic programming 71
3.4 Numerical investigation 72
3.5 Conclusion 75

Appendix **77**
3.A Transition parameters 77
3.B Parallel computing algorithm 79

Bibliography **81**

CONCLUSION **83**

LIST OF PAPERS

1. Hatem Ben-Ameur, Malek Ben-Abdellatif and Bruno Rémillard. Dynamic programming and parallel computing for valuing two-dimensional American-style options. Les Cahiers du GERAD G-2016-48, HEC Montréal, Montreal, QC, Canada, June 2016. Submitted.

2. Hatem Ben-Ameur, Malek Ben-Abdellatif and Bruno Rémillard. A two-factor structural model for valuing corporate securities. Les Cahiers du GERAD G-2016-119, HEC Montréal, Montreal, QC, Canada, December 2016.

3. Hatem Ben-Ameur, Malek Ben-Abdellatif and Bruno Rémillard. A structural model for valuing exchangeable bonds. *To appear in* Les Cahiers du GERAD, january 2017.

LIST OF TABLES

1.1 European put-on-min options DP vs. Boyle (1988) 15
1.2 American call-on-min options DP vs. Detemple et al. (2003) . 16
1.3 American put-on-min options DP vs. Boyle (1988) 18
1.4 American put-on-min options – DP vs. Rogers (2002), Jin et al. (2007), and Hartley (2000) 18
1.5 American call-on-max options – DP vs Broadie and Glasserman (1997) and Jin et al. (2007) 19
1.6 American call-on-max options – DP vs Raymar and Zwecher (1997) 20

LIST OF FIGURES

1.1 Illustration of the bilinear interpolation 9

1.2 Two-dimensional dynamic programming steps 10

1.3 Parallel computing using MPI 11

1.4 Dynamic programming tasks' parallelization 12

1.5 Briarée's architecture 14

1.6 Optimal policy for a put-on-min option 17

2.1 Credit spreads for an 8% bond for different leverage ratios. The parameters used are $r = 0.04, \alpha = 1, \beta = 0.06, \sigma_r = 0.03, \rho = -0.25, \sigma_v = 0.2, w = 0$, and $r^c = 0$. 40

2.2 Credit spreads for an 8% bond for different leverage ratios. The parameters used are $r = 0.04, \alpha = 1, \beta = 0.06, \sigma_r = 0.03, \rho = -0.25, \sigma_v = 0.2, w = 0.3$, and $r^c = 0.35$. 41

2.3 Credit spreads for an 8% bond for different values of r. The parameters used are $\alpha = 1, \beta = 0.06, \sigma_r = 0.03, \rho = -0.25, \sigma_v = 0.2, w = 0.3, r^c = 0.35$, and leverage ratio $= 0.5$. 42

2.4 Credit spreads for an 8% bond for different values of ρ. The parameters used are $r = 0.04, \alpha = 1, \beta = 0.06, \sigma_r = 0.03, \sigma_v = 0.2, w = 0.3, r^c = 0.35$, and leverage ratio $= 0.5$. 42

2.5 Credit spreads for an 8% bond for different values of σ_v. The parameters used are $r = 0.04, \alpha = 1, \beta = 0.06, \sigma_r = 0.03, \rho = -0.25, w = 0.3, r^c = 0.35$, and leverage ratio $= 0.5$. 43

3.1 Exchangeable debt value as a function of the shares' value and the firm's assets value. The parameters used are $r = 0.04, \delta_1 = 0.3, \sigma_V = 02, \delta_2 = 0, \sigma_S = 0.3, \rho = 0, w = 0.2$, and $r^c = 0.35$. . . 73

3.2 Exchangeable debt value as a function of the shares' value as the assets's volatility is changed. The parameters used are $r = 0.04, V_0 = 1.5, \delta_1 = 0.3, \delta_2 = 0, \sigma_s = 0.3, \rho = 0, w = 0.2$, and $r^c = 0.35$. 74
3.3 Exchangeable debt value as a function of the shares' value as the shares' volatility is changed. The parameters used are $r = 0.04, V_0 = 1.5, \delta_1 = 0.3, \sigma_V = 0.2, \delta_2 = 0, \rho = 0, w = 0.2$, and $r^c = 0.35$. 74
3.4 Exchangeable debt value as a function of the shares' value as the correlation is changed. The parameters used are $r = 0.04, \sigma_V = 0.2, \delta 1 = 0.3, V_0 = 1.5, \sigma_S = 0.3, \delta_2 = 0, w = 0.2$, and $r^c = 0.35$. 75

LIST OF APPENDICES

1.A Closed-form solutions for European options 22

 1.A.1 Exchange option 22

 1.A.2 Call-on-max option 22

 1.A.3 Put-on-min option 23

1.B Transition parameters 24

2.A Forward measure 46

2.B Transitions parameters 48

2.C Parallel computing algorithm 50

3.A Transitions parameters 77

3.B Parallel computing algorithm 79

Pour ma raison d'être, Lilia
et la source de ma force, Alya.

ACKNOWLEDGMENTS

Je tiens à exprimer mes plus grands remerciements et ma plus profonde reconnaissance à mes professeurs et directeurs de recherche Hatem Ben Ameur et Bruno Rémillard pour leur encadrement, leur patience, leur soutien et leur confiance. Grâce à eux j'ai beaucoup appris et j'ai pu mener à terme ce projet.

Je remercie également mes professeurs, mes collègues ainsi que le personnel administratif du département des sciences de la décision pour leur bonnes paroles et leurs encouragements. En particulier, merci à Diane pour son soutien et sa bienveillance.

J'exprime surtout ma plus grande gratitude envers mes parents *Mohamed* et *Wided* pour tous les sacrifices qu'ils ont faits pour moi ainsi que pour leur soutien inconditionnel, leur confiance infinie et leurs encouragements incessants. Vous avez toujours été une source d'inspiration pour moi et je suis comblée de voir cette fiéreté dans vos yeux.

Je remercie également mon frère *Kamel* et ma soeur *Boutheina* ainsi que ma belle soeur *Amira* et tous les membres de ma famille et mes amis qui m'ont soutenue et encouragée tout au long de mon parcours.

Une pensée spéciale à une personne d'exception dans ma vie, ma meilleure amie, ma soeur de coeur, ma combattante de parcours *Rim*, sans qui cette aventure n'aurait pas eu le même goût. Merci d'avoir toujours été là pour moi, de m'avoir toujours poussée à avancer, et d'avoir toujours cru en moi, même dans les moments les plus difficiles. Je garderai de ces années le résonnement de nos rires au milieu des doutes, les moments d'espoir au milieu des incertitudes et notre persévérance malgré tout. Merci d'avoir croisé mon chemin!

INTRODUCTION

Modeling complex situations in finance usually requires several factors to be taken into account to most accurately reproduce the observed effects. In particular, multi-factor models are popular in financial engineering, for instance valuing derivative contracts written on several underlying assets. Multi-assets options are used for hedging market risk related to financial assets, natural resources, and commodities. Examples include rainbow, quanto, and basket options. In credit risk modeling, multi-factor models are also used since credit risk usually depends on firm-specific factors but also on economy-specific factors such as interest rate or exchange rate for foreign companies. Other examples include specific financial products naturally involving more than one factor, such as reverse convertible bonds or exchangeable bonds.

For most cases, when increasing the problem dimension, closed-form solutions are no longer available, and using numerical procedures becomes inevitable. In this thesis, we focus on dynamic programming coupled with finite elements. This methodology performs well for one-dimensional problems and has proved its efficiency, especially for American-style contracts. The extension of dynamic programming to higher dimensions is natural, as long as the transition parameters can be computed in closed form. Since execution times become too long, parallel computing is used to avoid this problem and enhance efficiency. In fact, several computational tasks in dynamic programming, including the calculation of transition parameters and the evaluation of the value function on the grid, are not sequential and thus can be parallelized. These independent tasks are then executed simultaneously, which makes it possible to reduce the calculation times drastically, going from more than 24 hours of calculations to a few seconds.

In this thesis, we propose a two-dimensional dynamic program coupled with finite elements where we use parallel computing to enhance efficiency. The three essays propose different applications of the use of our methodology in a two-dimensional setting.

The first essay proposes a dynamic program for valuing two-dimensional American-style options. Our numerical experiments show convergence and efficiency, and puts forward our methodology as a competitive alternative in comparison to traditional approaches. In addition, our model is flexible because it accommodates a large family of option contracts signed on two underlying assets that move according to a lognormal vector process. The same procedure can be adapted to accommodate a larger family of derivative contracts and state-process dynamics.

The second essay presents a general structural model for valuing risky corporate debts when the interest rates are stochastic. Our two-factor model takes into account both credit default risk and interest rate risk. The two factors are the firm's assets value that moves according to a lognormal process, and the short-term interest rate that follows a mean-reverting Gaussian process. Our model is flexible as it accommodates an arbitrary debt structure, multiple seniority classes, tax benefits and bankruptcy costs. Our results are consistent with empirical evidence stipulating that modeling interest rate risk can explain observed credit spreads, in particular, credit spread differences for similarly rated bonds.

In the third essay, we focus on the valuation of exchangeable bonds which are debts that are convertible into shares of a firm's equity other than the bond's issuer. These structured bonds gained popularity in recent years and are discussed widely in corporate finance but less attention has been paid to their evaluation, on which we focus in this paper. We propose a two-dimensional structural model where the issuer's assets value and the underlying equity shares are the state variables and follow a lognormal vector process. Our model accommodates an arbitrary debt structure including an exchangeable bond, tax benefits as well as bankruptcy costs. Our results highlight the main characteristics of exchangeable bonds.

CHAPTER 1

DYNAMIC PROGRAMMING AND PARALLEL COMPUTING FOR VALUING TWO-DIMENSIONAL AMERICAN-STYLE OPTIONS

Malek Ben-Abdellatif,[1] Hatem Ben Ameur,[2] Bruno Rémillard[3]

Abstract

We propose a dynamic program coupled with finite elements for valuing two-dimensional American-style options. To speed-up our procedure, we use parallel computing at every step of the recursion. Our model is flexible because it accommodates a large family of option contracts signed on two underlying assets that move according to a lognormal vector process. The same procedure can be adapted to accommodate a larger family of derivative contracts and state-process dynamics. Our numerical experiments show convergence and efficiency, positioning our method as a viable alternative to traditional methodologies based on trees, finite differences, and Monte Carlo simulation.

Keywords: American options; Two-dimensional state spaces; Dynamic programming; Finite elements; Parallel computing.

1.1 Introduction

We propose a dynamic program for valuing two-dimensional American-style options. Parallel computing is used to reduce its running time and enhance its efficiency. Two-dimensional options are traded over the counter (OTC). Examples

1. Malek Ben-Abdellatif is a Ph.D. student at HEC Montréal.

2. Hatem Ben-Ameur is a Professor at HEC Montréal in the Department of Decision Sciences, and member of the GERAD.

3. Bruno Rémillard is a Professor at HEC Montréal in the Department of Decision Sciences, and member of the CRM and GERAD.

Partial funding in support of this work was provided by the Natural Sciences and Engineering Research Council of Canada, the Fonds pour la formation de chercheurs et l'aide à la recherche du Gouvernement du Québec, and the Institut de Finance Mathématique du Québec.

include rainbow, quanto, and basket options. According to the Bank of International Settlements (BIS), the gross market value of total OTC options was 438 billion US dollars in 2015. These options are used for hedging market risk related to financial assets, natural resources, and commodities.

American-style options cannot be evaluated in closed form and must be approximated in some way. The literature discusses several methodologies based on Monte Carlo simulation, trees, and finite differences. This holds for one-dimensional as well as multidimensional state spaces.

The multidimensional lattice approach assumes a discrete model that usually converges to a continuous counterpart. It runs in two steps. The first step, which is the most challenging, is forward and used for the lattice construction (the overall state space). The second step is backward and is used for evaluation purposes (Boyle, 1988, Boyle et al., 1989, Kamrad and Ritchken, 1991).

Finite differences (FD) is a backward methodology that assumes state and time discretizations. At each step of the recursion, one solves a partial differential equation with boundary conditions, which characterizes the option's value. While Dockendorf and Paxson (2015) use FD for valuing multidimensional European-style options, Hartley (2000) and Berridge and Schumacher (2008) use FD for valuing their American counterparts.

Simulation-based methodologies run in two steps. The first step is forward, and involves simulating a random sample of the underlying assets' trajectories, which can be seen as a random lattice. The second step, which is the most challenging, is backward and evaluates the option contract and identifies its optimal exercise strategy. At any given decision date, the simulated trajectories do not intersect (almost surely), which weakens the evaluation step via the fundamental theorem of asset pricing in no-arbitrage markets. The solution proposed in the literature is threefold: bundling methods (Tilley, 1993, Barraquand and Martineau, 1995, Boyle et al., 1997, Raymar and Zwecher, 1997, Jin et al., 2007, 2013, Broadie and Glasserman, 1997, Bally and Pages, 2003a,b, Bally and Printems, 2005), regression and/or global approximations (Carriere, 1996, Tsitsiklis and Van Roy, 1999, Longstaff and

Schwartz, 2001, Broadie and Glasserman, 1997), and duality methods (Haugh and Kogan, 2004, Rogers, 2002, Andersen and Broadie, 2004, Del Moral et al., 2012).

Dynamic programming (DP) coupled with finite elements has been used with success for valuing one-dimensional American-style options in pure-diffusion models (Ben-Ameur et al., 2002) and jump-diffusion models (Ben-Ameur et al., 2016). Design-wise, one-dimensional DP can be naturally extended to higher dimensions and combined with parallel computing in order to reduce computation time.

DP starts running at the option's maturity, where the option's value function is known. The option's value is computed on a given grid of points. Then, DP alternates between interpolation at step $n+1$ and evaluation at step n, and moves backward from maturity down to the origin. At each step of the recursion, DP acts as follows: one first uses a piecewise polynomial and interpolates the option's value function from the grid points to the overall state space; then one uses no-arbitrage pricing and approximates the option's value function at the previous step on (possibly) the same grid points. These last computational tasks are not sequential and, thus, can be parallelized.

This paper is organized as follows. While Section 1.2 presents the model, Section 1.3 describes our dynamic program and discusses parallel computing. Section 1.4 is a numerical investigation, and Section 1.5 concludes.

1.2 The model

We consider a frictionless market in which two stocks, S^1 and S^2, are traded continuously and move according to a bivariate log-normal process. The risk-free rate, r, is assumed to be constant. This market is known to be arbitrage free and complete. Thus, there exists a unique risk-neutral probability measure $\mathbb{Q}$ under which the state process (S^1, S^2) moves according to the following stochastic differential equation:

$$\frac{dS_t^i}{S_t^i} = (r - d_i)dt + \sigma_i dW_t^i, \quad \text{for } i = 1, 2, \tag{1.1}$$

where d_i is the continuous dividend rate of stock i, σ_i is its log-return volatility, and (W^1, W^2) is a bivariate correlated Brownian motion with

$$\mathrm{Cor}(W_t^1, W_t^2) = \rho, \quad \text{for all } t > 0.$$

The solution of (1.1) can be written as

$$S_u^i = S_t^i \exp\left[\left(r - d_i - \frac{\sigma_i^2}{2}\right)(u-t) + \sigma_i\left(W_u^i - W_t^i\right)\right], \text{for } 0 \leq t \leq u. \tag{1.2}$$

An American option on (S^1, S^2) with maturity T is defined by its cash-flow process, $\kappa(t,x,y) \geq 0$, for $0 \leq t \leq T$, $x > 0$, $y > 0$, where $x = S_t^1$ and $y = S_t^2$. This is the option's value under exercise. Its European counterpart is characterized by

$$\kappa(t,x,y) = 0, \quad \text{for } 0 \leq t < T.$$

Examples include the exchange option:

$$\kappa(t,x,y) = \max(x-y, 0),$$

the call-on-max option:

$$\kappa(t,x,y) = \max(\max(x,y) - K, 0),$$

and the put-on-min option:

$$\kappa(t,x,y) = \max(K - \min(x,y), 0),$$

where K is the the option's strike price. The exchange option gives the option holder the right to exchange S^2 for S^1; the call-on-max option gives the right to purchase the higher-priced asset at the strike price K; and the put-on-min gives the right to sell the lower-priced asset at the strike K.

This setting is described in detail in Stulz (1982) and Johnson (1987) where closed-form solutions for the above-mentioned European options are given. We report them in Appendix 1.A.

1.3 Dynamic programming and parallel computing

First, we describe the dynamic programming approach where the assets are modeled by a general Markov process. Then, we present the use of parallel computing to improve efficiency.

1.3.1 Dynamic programming

We consider a two-dimensional Bermudan option, characterized by its exercise value $v_t^e(x,y) = \kappa(t,x,y)$ and $N+1$ exercise opportunities, $t_0 = 0, t_1, \ldots, t_N = T$.

Let $\mathscr{G}$ be a set of grid points $\{(a_1,b_1),(a_1,b_2),\ldots,(a_p,b_q)\}$ such that $\max(\Delta a_k, \Delta b_l) \to 0$ and $\mathbb{Q}[(S_t^1, S_t^2) \in [a_p,\infty) \times \mathbb{R}_+^* \cup \mathbb{R}_+^* \times [b_q,\infty)] \to 0$, when p and $q \to \infty$. Let $a_0 = b_0 = 0$ and $a_{p+1} = b_{q+1} = \infty$. The rectangle $[a_i, a_{i+1}) \times [b_j, b_{j+1})$ is designated by R_{ij}.

For simplicity, assume that the state process (S^1, S^2) is Markov and homogeneous, $t_{n+1} - t_n = \Delta t$ a positive constant, the grid points $\mathscr{G}$ fixed along the recursion, and $\kappa(t,x,y) = \kappa(x,y)$. To reach the pure American option, let $\Delta t \to 0$.

Define the transition tables T^{00}, T^{10}, T^{01}, and T^{11} as follows:

$$\begin{aligned} T_{klij}^{\nu\mu} &= \mathbb{E}^*\Big[(S_{t_{n+1}}^1)^\nu (S_{t_{n+1}}^2)^\mu \mathbb{I}\left((S_{t_{n+1}}^1, S_{t_{n+1}}^2) \in R_{ij}\right) \mid \\ &\qquad (S_{t_n}^1, S_{t_n}^2) = (a_k, b_l)\Big], \quad \text{for } \nu \text{ and } \mu \in \{0,1\}. \end{aligned} \tag{1.3}$$

For example, T_{klij}^{00} represents the transition probability that the Markov process (S^1, S^2) moves from (a_k, b_l) at t_n and visits the rectangle R_{ij} at t_{n+1}. These transition parameters, which are at the heart of the dynamic-programming approach, can be considered a fixed cost as long as the Markov state process is homogeneous, $t_{n+1} - t_n$ is a positive constant, and the grid points $\mathscr{G}$ do not depend on time. We derive and provide closed-form solutions for them in Appendix 1.B.

Assume that an approximation of the option's value function is available at a future decision date t_{n+1} on $\mathscr{G}$, indicated by $\tilde{v}_{n+1}(a_k, b_l)$, for $k = 1, \ldots, p$ and $l = 1, \ldots, q$. This is not really a strong assumption since the option's value function is known at maturity in closed form, that is, $\tilde{v}_N = v_N = v_N^e$. DP acts as follows:

1. Use a bilinear piecewise polynomial (as illustrated in Figure 1.1) and interpolate the option's value function $\tilde{v}_{n+1}$ at t_{n+1} from $\mathscr{G}$ to the overall state space $[0, \infty)^2$ by setting

$$\begin{aligned} \hat{v}_{n+1}(x,y) &= \sum_{i=0}^{p}\sum_{j=0}^{q}\left(\alpha_{ij}^{n+1} + \beta_{ij}^{n+1}x + \gamma_{ij}^{n+1}y + \delta_{ij}^{n+1}xy\right) \times \\ &\quad \mathbb{I}\left((x,y) \in R_{ij}\right), \end{aligned} \tag{1.4}$$

   where the local coefficients α_{ij}^{n+1}, β_{ij}^{n+1}, γ_{ij}^{n+1}, and δ_{ij}^{n+1}, for $i = 1, \ldots, p-1$ and $j = 1, \ldots, q-1$, are obtained in closed form from the system of linear equations:

$$\begin{cases} \hat{v}_{n+1}(a_i, b_j) = \tilde{v}_{n+1}(a_i, b_j) \\ \hat{v}_{n+1}(a_{i+1}, b_j) = \tilde{v}_{n+1}(a_{i+1}, b_j) \\ \hat{v}_{n+1}(a_i, b_{j+1}) = \tilde{v}_{n+1}(a_i, b_{j+1}) \\ \hat{v}_{n+1}(a_{i+1}, b_{j+1}) = \tilde{v}_{n+1}(a_{i+1}, b_{j+1}) \end{cases} \tag{1.5}$$

   and the rest of them are set to their adjacent counterparts;

2. Use non-arbitrage pricing and approximate the option's holding value function at t_n on $\mathscr{G}$:

$$\begin{aligned} \tilde{v}_n^h(a_k, b_l) &= \mathbb{E}^*\left[e^{-r\Delta t}\hat{v}_{n+1}(S_{t_{n+1}}^1, S_{t_{n+1}}^2) \mid (S_{t_n}^1, S_{t_n}^2) = (a_k, b_l)\right] \\ &= e^{-r\Delta t}\sum_{i,j}\left(\alpha_{ij}^{n+1}T_{klij}^{00} + \beta_{ij}^{n+1}T_{klij}^{10} + \gamma_{ij}^{n+1}T_{klij}^{01} + \delta_{ij}^{n+1}T_{klij}^{11}\right); \end{aligned} \tag{1.6}$$

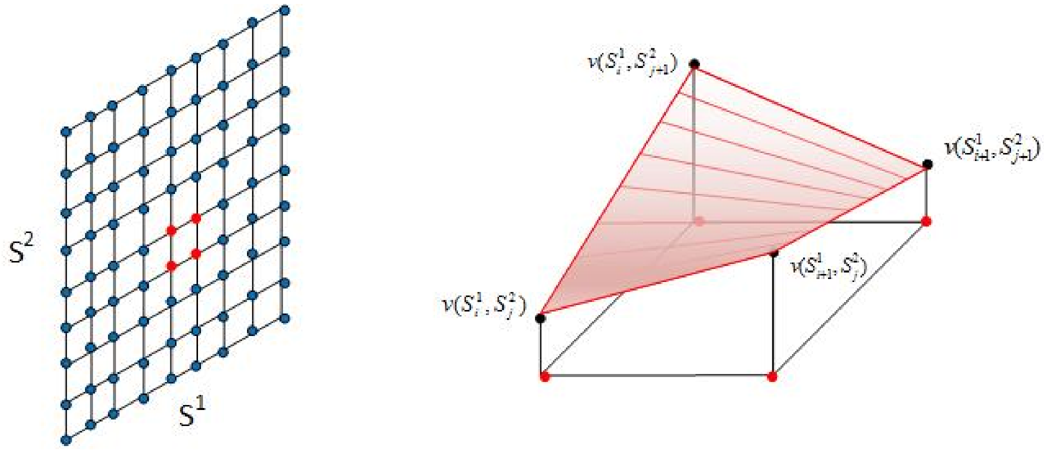

Figure 1.1: Illustration of the bilinear interpolation

3. Approximate the option's value function at t_n on $\mathscr{G}$:

$$\tilde{v}_n(a_k, b_l) \quad = \quad \max(v_n^e(a_k, b_l), \tilde{v}_n^h(a_k, b_l)); \tag{1.7}$$

4. Go to step 1 and repeat until $n = 0$.

These steps are illustrated in Figure 1.2. Eq. (3.2) splits the option's holding value into two parts: the local coefficients are related to the option contract and the transition parameters to the dynamics of the state process. All in all, the option's holding value is a sum of local future-value pieces, times their associated transition parameters, discounted back at the risk free-rate. The same equation shows that DP assumes a space discretization, but not a time discretization, and does respect the true dynamics of the state process as long as the transition parameters in eq. (3) are known in closed form. Finally, eq. (4) shows that DP ends up with an interpolation of $v_0(x, y)$, defined on the overall state space. Thus, the first and second derivatives of $v_0(x, y)$ with respect to x and y then become available, among other sensitivity coefficients. In particular, the deltas are obtained, at each time step, as follows:

$$\frac{\partial \hat{v}_{n+1}(x, y)}{\partial x} = \left(\beta_{ij}^{n+1} + \delta_{ij}^{n+1} y\right) \mathbb{I}\left((x, y) \in R_{ij}\right),$$

and

$$\frac{\partial \hat{v}_{n+1}(x,y)}{\partial y} = \left(\gamma_{ij}^{n+1} + \delta_{ij}^{n+1} x\right) \mathbb{I}\left((x,y) \in R_{ij}\right),$$

where the couple (x,y) is assumed to be inside the rectangle R_{ij}.

Higher-order two-dimensional piecewise approximations are more accurate but require a higher computing time, and vice versa. We find that the bilinear piecewise interpolation of eq. (1.4) is an acceptable compromise.

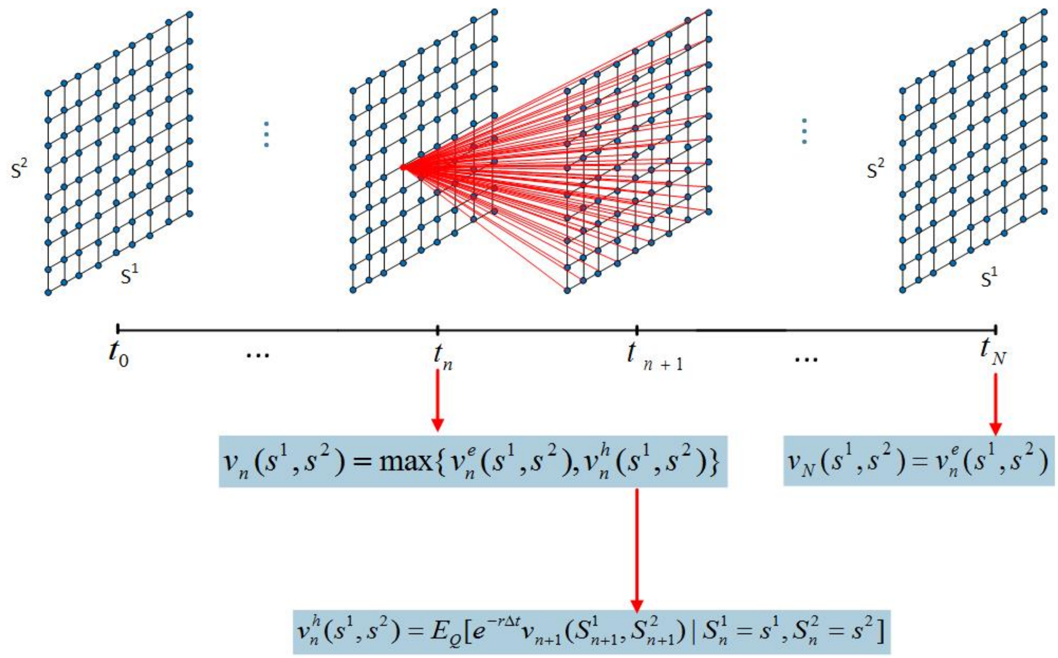

Figure 1.2: Two-dimensional dynamic programming steps

1.3.2 Parallel computing

Parallel computing uses multiple central processing units (CPUs) simultaneously to speed-up complex computations. For C programming, used herein to achieve our numerical experiments, there are two libraries used for parallel computing: MPI and OpenMP.

The Message Passing Interface (MPI) library allows the computing process to exchange information between the running CPU environments in order to achieve a given job. Each CPU has access to a certain memory space. MPI requires case-sensitive programming changes from the serial code to its parallel version.

Parallel computing can also run when all CPUs share the same memory space. Open Multi Processing (OpenMP) is a library that allows one to implement parallel computing with a minimal change to the serial code. However, shared-memory supercomputers are extremely expensive, and thus somewhat inaccessible.

MPI and OpenMP are compatible with Fortran and C languages. Parallel computing is also feasible under other software packages, e.g., Graphics Processing Unit (GPU) for Matlab and R.

Parallel computing using the MPI library acts as described in Figure 1.3. The code usually starts with a serial code until we have independent tasks that can be executed simultaneously. We can then call the parallel computing environment and divide the work between the available CPUs. We choose a master CPU, which will be responsible for collecting and assembling the total work. The other CPUs are called slaves. When these tasks are done, we proceed with the exchange of information between the master and the slaves, which ends the parallel part of the code. We can then go back to serial code. The information transfers have to be minimal for efficiency purposes.

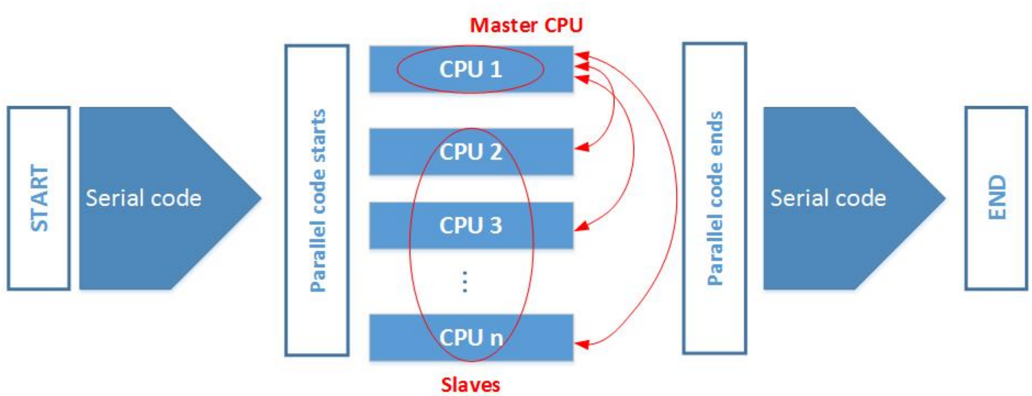

Figure 1.3: Parallel computing using MPI

The easiest way to parallelize DP is to submit the computation tasks associated to a given grid point (a_k, b_l), for $k = 1, \dots, p$ and $l = 1, \dots, q$, to a single CPU. Our parallel code acts as follows.

1. This single CPU computes once and locally stores the overall grid points (a_i, b_j) and the exercise values $\kappa(a_i, b_j)$, for $i = 1, \ldots, p$ and $j = 1, \ldots, q$.
2. Following eq. (3), it also computes once and locally stores the $4 \times (p+1)(q+1)$ transition parameters T^{00}_{klij}, T^{10}_{klij}, T^{01}_{klij}, and T^{11}_{klij}, for $i = 0, \ldots, p$ and $j = 0, \ldots, q$.
3. Following eq. (4)-(5), it computes and stores at step $n+1$ the local coefficients α^{n+1}_{ij}, β^{n+1}_{ij}, γ^{n+1}_{ij}, and δ^{n+1}_{ij}, for $i = 0, \ldots, p$ and $j = 0, \ldots, q$.
4. Following eq. (6)-(7), it computes and stores at step n the option's holding value $\tilde{v}^h_n(a_k, b_l)$ then the overall value $\tilde{v}_n(a_k, b_l)$.
5. The same CPU exports $\tilde{v}_n(a_k, b_l)$ to a selected CPU, the so-called master CPU.
6. The master CPU collects $\tilde{v}_n(a_k, b_l)$, for $k = 1, \ldots, p$ and $l = 1, \ldots, q$, and sends them back to all running CPUs.
7. Go to step 3 and repeat until $n = 0$.

Since the number of CPUs available to the analyst is usually less than the grid size pq, we submit the same number of grid points to each CPU as described in figure 1.4. Fixing this number for each grid size pq is a question of efficiency. Assume the same program is run twice with n and kn CPUs, where n and $k \in \mathbb{N}^*$.

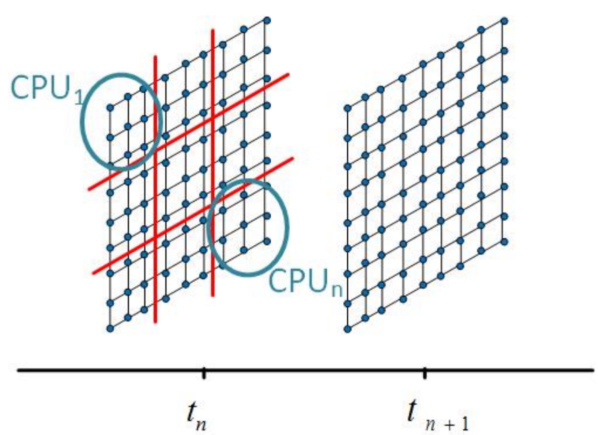

Figure 1.4: Dynamic programming tasks' parallelization

Let τ_1 and τ_2 be the computing times of the first and second run, respectively. In the best case scenario, the expected computing time declines by the same factor k, that is,

$$E[\tau_2] = \frac{E[\tau_1]}{k},$$

which results in a relative efficiency ratio

$$\frac{E[\tau_1]/E[\tau_2]}{k} = 1.$$

In fact, this ratio is usually lower than one, since the running CPUs exchange some information during the computing process, as in steps 5-6, and the parallel code behaves partially as the serial code, as in step 1. A relative efficiency ratio higher than 75% is highly desirable.

We use the supercomputer Briarée managed by Calcul Québec and Compute Canada[4]; it is equipped with 8064 CPUs (cores). These 8064 cores are divided in 672 computing nodes, each equipped with two six-core processors running at a speed of 2.667 GHz. Thus, each computing node includes 12 cores. The number of computing nodes, $\bar{n}$, called for parallel computing must be specified by the programmer ($\bar{n} \leq 672$), which results in $12 \times \bar{n}$ cores. Briarée has a total memory space of 26.72 TB, split between the computing nodes. Given the architecture of Briarée's hardware (Figure 1.5), the number of grid points submitted to each core is

$$\frac{pq}{12 \times \bar{n}} \in \mathbb{N}^*.$$

The code lines are written in C and compiled with GCC. We use the MPI library to access parallel computing.

4. The operation of this supercomputer is funded by the Canada Foundation for Innovation (CFI), Ministère de l'Économie, de la Science et de l'Innovation du Québec (MESI) and the Fonds de recherche du Québec - Nature et technologies (FRQ-NT).

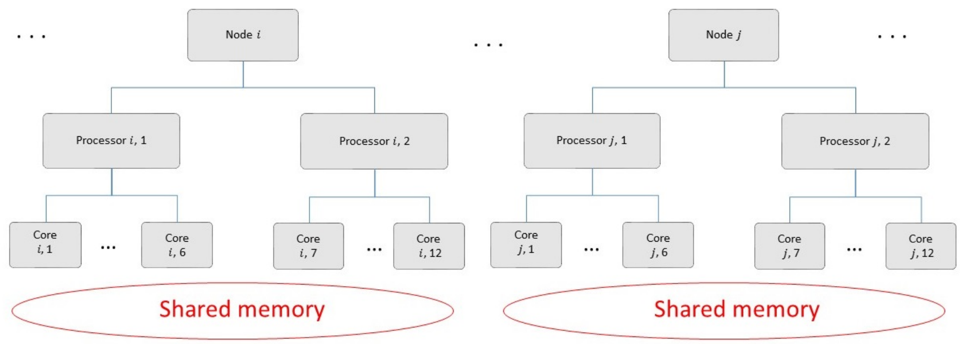

Figure 1.5: Briarée's architecture

1.4 Numerical investigation

1.4.1 European options

Table 1.1 compares DP to Boyle (1988), who uses a two-dimensional trinomial tree for valuing European put-on-min options. The closed-form solution for this contract is given in Stulz (1982) (see also Appendix 1.A). Set $S_0^1 = S_0^2 = 40$, $d_1 = d_2 = 0$, $\sigma_1 = 0.2$, $\sigma_2 = 0.3$, $\rho = 0.5$, $r = 5\%$ (effective) $\equiv 0.04879$ (continuously compounded), $T = 7$ months $\equiv 0.58333$ years. Following the constraints of Calcul Québec and Compute Canada, we select $\bar{n}$ as the highest integer lower than 200 that ensures an efficiency ratio higher than 75%, where $pq/(12\bar{n}) \in \mathbb{N}^*$.

As explained in Section 3, DP does not need a time discretization. For comparison purposes, however, we run DP with the same number of time steps as assumed in Boyle (1988). As expected, when the number of time steps is low, DP behaves almost perfectly, whereas the binomial tree is less accurate. For high number of time steps, the binomial tree converges and becomes as accurate as DP. It is worth noticing that DP over-evaluate the true value and approaches it as the grid size increases. This can be explained by the fact that the bilinear interpolation always over-approximates the true value function as the latter is convex.

Boyle (1988) does not report his computing times. Each DP's CPU time (in seconds) can be split into a fixed cost, associated to the transition parameters, and

	DP with a grid size pq				
	72^2	144^2	300^2	Boyle	Closed form
$K = 35$	1.411	1.392	1.388	1.425	1.387
40	3.837	3.805	3.800	3.778	3.798
45	7.543	7.508	7.501	7.475	7.500
$12\times\bar{n}$	576	1728	1800		
Total CPU	0.93	5.40	34.48		
Linear CPU	0.65	4.11	11.99	10 time steps	
$K = 35$	1.504	1.410	1.391	1.392	1.387
40	3.970	3.832	3.805	3.795	3.798
45	7.694	7.537	7.507	7.499	7.500
$12\times\bar{n}$	576	1728	1800		
Total CPU	2.14	10.29	67.01		
Linear CPU	1.83	8.85	46.91	50 time steps	

Table 1.1: European put-on-min options
DP vs. Boyle (1988)

a linear cost, associated to the backward recursion. Our numerical experiments show that the fixed cost accounts for an important portion of the total CPU time. The relevant DP's computing time is the linear CPU time, since the transition parameters can be computed only once or twice a day, following the model-estimation step.

1.4.2 American options

We compare DP to alternative methodologies for valuing American-style options, that is, the lattice approach, finite differences, and Monte Carlo simulation, among other ad hoc procedures.

		DP with a grid size pq				
S_0^2	S_0^1	72^2	144^2	300^2	Lower bound	Upper bound
90	90	0.893	0.799	0.782	0.63	0.97
	100	1.691	1.560	1.536	1.27	1.88
	110	2.526	2.364	2.335	1.92	2.77
	120	3.177	2.984	2.950	2.49	3.37
100	100	3.392	3.233	3.205	2.62	3.98
	110	5.341	5.174	5.145	4.26	6.36
	120	6.680	6.608	6.576	5.51	7.58
	130	7.639	7.402	7.361	6.36	7.95
110	110	9.455	9.362	9.348	10.00	13.66
	120	12.225	12.118	12.098	10.37	14.50
	130	13.511	13.335	13.304	11.58	14.53
	140	14.147	13.912	13.869	12.51	14.44
$12\times\bar{n}$		576	1728	1800		
Total CPU		2.93	15.25	81.88		
Linear CPU		2.89	15.05	78.76	100 time steps	

Table 1.2: American call-on-min options
DP vs. Detemple et al. (2003)

Detemple et al. (2003) consider an American call-on-min option. They propose several approximations of the option's exercise frontier at each decision date, which results in a lower and an upper bound for the option's value. These bounds are then estimated by Monte Carlo simulation of size $50,000$. Table 1.2 reports DP values versus Detemple et al. (2003). The parameters are $K = 100$, $d_1 = d_2 = 0.05$, $\rho = 0$, $\sigma_1 = \sigma_2 = 0.2$, $T = 1$, and $r = 0.06$. DP values are always between their lower and upper counterparts.

Table 1.3 compares DP to Boyle's (1988) trinomial tree. The parameters are given in Section 4.1. All in all, DP values are close to Boyle's (1988) values. Consistent with the analysis of Section 4.1, the gap between the competing approximations is larger when the number of exercise opportunities is low; DP is expected to be more accurate. In addition, figure 1.6 plots the exercise region of a put-on-min option at date 4 for the same parameter as table 1.3 with strike price $K = 35$ and 10 decision dates.

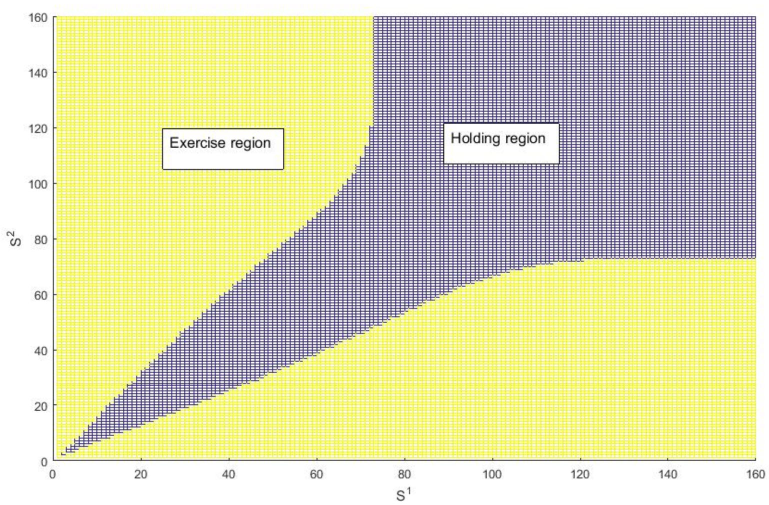

Figure 1.6: Optimal policy for a put-on-min option

While Monte Carlo simulation is combined with a dual approach by Rogers (2002), it is combined with a bundling approach by Jin et al. (2007). Their random samples are of size $10,000$ and $60,000$, respectively. We report their respective 95% confidence intervals. Hartley (2000) uses finite differences. The parameters are $K = 100$, $d_1 = d_2 = 0$, $\rho = 0$, $\sigma_1 = \sigma_2 = 0.6$, $r = 0.06$, and $T = 0.5$.

	DP with a grid size pq			
	72^2	144^2	300^2	Boyle
$K = 35$	1.436	1.416	1.413	1.450
40	3.918	3.887	3.881	3.870
45	7.713	7.678	7.671	7.645
$12\times\bar{n}$	576	1728	1800	
Total CPU	1.01	5.79	36.63	
Linear CPU	0.67	4.51	13.75	10 time steps
$K = 35$	1.535	1.440	1.422	1.423
40	4.064	3.926	3.899	3.892
45	7.880	7.727	7.697	7.689
$12\times\bar{n}$	576	1728	1800	
Total CPU	1.96	9.94	66.45	
Linear CPU	1.71	8.46	46.43	50 time steps

Table 1.3: American put-on-min options
DP vs. Boyle (1988)

	DP with a grid size pq					
(S_0^1, S_0^2)	72^2	144^2	300^2	Rogers	Jin et al.	Hartley
(80, 80)	37.938	37.416	37.312	[37.35, 37.65]	[37.1000, 37.4022]	37.30
(80, 100)	32.775	32.205	32.091	[32.12, 32.26]	[31.8421, 32.1451]	32.08
(80, 120)	29.826	29.265	29.152	[29.18, 29.32]	[28.8860, 29.2434]	29.14
(100, 100)	25.889	25.205	25.066	[24.93, 25.23]	[24.8296, 25.1576]	25.06
(100, 120)	21.779	21.067	20.920	[20.89, 21.09]	[20.6850, 20.9932]	20.91
(120,120)	16.864	16.102	15.942	[15.99, 16.19]	[15.6737, 16.0017]	15.92
$12\times\bar{n}$	576	1728	1800			
Total CPU	1.66	8.70	46.72	180	24	
Linear CPU	1.60	8.41	43.58	51 time steps		

Table 1.4: American put-on-min options – DP vs. Rogers (2002), Jin et al. (2007), and Hartley (2000)

DP values, obtained with $p = q = 300$, almost always belong to their associated 95% confidence intervals and compare extremely well with Hartley's (2000) values, which are described by Rogers (2002) as extremely accurate.

Table 1.5 compares DP to the algorithm of Jin et al. (2007) for Bermudan call on max options on two assets (JTS). BG stands for Broadie and Glasserman (1997) who propose two Monte Carlo estimators, one biased low and the other biased hight. We report a 95% confidence intervals (CI) for the true values. Set $K = 100$, $d_1 = d_2 = 0.1$, $\rho = 0.3$, $\sigma_1 = \sigma_2 = 0.2$, $T = 1$ (year), and $r = 0.05$. All procedures run with four evenly exercise dates. True values are reported from Broadie and Glasserman (1997) and are approximated using Kamrad and Ritchken (1991). Their are based on a two-point Richardson extrapolation of the lattice value with 600 time steps and with a 1200 time steps.

	DP with a grid size p					
(S_0^1, S_0^2)	72	144	300	Broadie and Glasserman	Jin et al.	True
(70, 70)	0.244	0.239	0.238	[0.231, 0.266]	[0.2224, 0.2436]	0.237
(80, 80)	1.296	1.273	1.269	[1.191, 1.287]	[1.2159, 1.2865]	1.259
(90, 90)	4.179	4.119	4.108	[3.923, 4.216]	[4.0141, 4.1365]	4.077
(100, 100)	9.522	9.442	9.427	[9.046, 9.673]	[9.2672, 9.4302]	9.361
(110, 110)	17.216	17.072	17.040	[16.516, 17.504]	[16.8415, 17.0037]	16.924
(120, 120)	26.449	26.237	26.161	[25.471, 26.643]	[25.8637, 26.1471]	25.980
(130, 130)	36.537	36.176	36.033	[35.161, 36.646]	[35.6321, 35.9155]	35.763
$12\times\bar{n}$	576	1728	1800			
Total CPU	1.28	7.35	118.78		1.06	
Linear CPU	0.45	2.55	14.45		4 time steps	

Table 1.5: American call-on-max options – DP vs Broadie and Glasserman (1997) and Jin et al. (2007)

Table 1.6 compares DP to Raymar and Zwecher (1997), who use a simulation based algorithm. Their procedure runs with 200.000 paths and 10 time steps and their prices are noted (RZ). Set $d_1 = d_2 = 0.1$, $\rho = 0.3$, $\sigma_1 = \sigma_2 = 0.2$, $T = 1$ (year)

(S_0^1, S_0^2)	K	DP with a grid size p: 72	144	300	Raymard and Zwecher	Binomial
(100, 110)	70	40.998	40.940	40.931	[40.852, 40.986]	40.930
	100	14.037	13.959	13.946	[13.923, 14.075]	13.945
	130	2.152	2.106	2.097	[2.173, 2.263]	2.093
(100, 100)	70	34.601	34.537	34.526	[34.453, 34.605]	34.525
	100	9.632	9.559	9.547	[9.542, 9.694]	9.543
	130	1.100	1.071	1.065	[1.125, 1.187]	1.062
(100, 90)	70	30.838	30.786	30.777	[30.698, 30.824]	30.776
	100	7.237	7.172	7.160	[7.169, 7.303]	7.158
	130	0.706	0.684	0.680	[0.732, 0.786]	0.677
$12\times\bar{n}$		576	1728	1800		
Total CPU		1.25	5.09	105		
Linear CPU		0.29	1.20	18.04	10 time steps	

Table 1.6: American call-on-max options – DP vs Raymar and Zwecher (1997)

and $r = 0.05$. Binomial prices are taken from Raymar and Zwecher (1997) and based on 10 time steps.

DP shows competitive CPU times. These could have been further drastically reduced with access to the overall hardware capabilities. These results reinforce DP as a viable alternative for valuing two-dimensional American options.

1.5 Conclusion

We propose a dynamic program coupled with piecewise bilinear approximations for valuing two-dimensional American options. We use parallel computing to speed up efficiency. This methodology presents two major advantages with respect to its competitors, giving that it assumes a space but not a time discretization, and a numerical but not a statistical error. Our investigation shows that DP competes well against its alternative methodologies in terms of accuracy. Although DP's CPU times are competitive, they can be further drastically reduced through access to the overall hardware capabilities.

This paper paves the way for a few useful extensions. The same DP procedure can accommodate a larger family of derivative contracts, such as Asian options and barrier options, and more complex state processes, such as two-dimensional jump diffusions and GARCH processes, as long as the transition parameters can be computed efficiently. The extension to higher dimensions is challenging but feasible. Monte Carlo simulation is certainly required for valuing option contracts in high state-space dimensions, but DP can firstly be combined with quasi–Monte Carlo simulation in moderate state-space dimensions, where the latter deterministic approach is known to be more efficient than the former random approach.

APPENDIX

1.A Closed-form solutions for European options

The closed-form solutions given below are taken from Stulz (1982).

1.A.1 Exchange option

The price of an European exchange option giving the right to exchange S^2 against S^1 at maturity date T, evaluated at date t, is given by

$$E(S_t^1, S_t^2, T-t) = S_t^1 e^{-d_1(T-t)}\Phi(d_+) - S_t^2 e^{-d_2(T-t)}\Phi(d_-),$$

where

$$\begin{aligned} d_{\pm} &= \frac{ln(S_t^1/S_t^2) - (d_1 - d_2 \pm \sigma^2/2)(T-t)}{\sigma\sqrt{(T-t)}}, \\ \sigma &= \sqrt{\sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho}, \end{aligned}$$

and $\Phi(\cdot)$ is the cumulative density function of the univariate standard normal distribution.

1.A.2 Call-on-max option

The price of an European call-on-max option with maturity date T and strike price K, evaluated at date t, is given by

$$\begin{aligned} C_{\max}(S_t^1, S_t^2, K, T-t) &= S_t^1 e^{-d_1(T-t)}\Phi(d_1^1, d_{12}, \rho_{12}) + \\ & S_t^2 e^{-d_2(T-t)}\Phi(d_1^2, d_{21}, \rho_{21}) - \\ & Ke^{-r(T-t)}(1 - \Phi(-d_2^1, -d_2^2, \rho)), \end{aligned}$$

where

$$
\begin{aligned}
\sigma &= \sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2, \\
d_{ij} &= \frac{log(S_t^i/S_t^j) + (d_j - d_i + \sigma^2/2)(T-t)}{\sigma\sqrt{(T-t)}}, \\
\rho_{ij} &= \frac{\sigma_i - \rho\sigma_j}{\sigma}, \\
d_2^i &= \frac{log(S_t^i/K) + (r - d_i - \sigma_i^2/2)(T-t)}{\sigma_i\sqrt{T-t}}, \\
d_1^i &= d_2^i + \sigma_i\sqrt{T-t},
\end{aligned}
$$

and $\Phi(\cdot,\cdot,\rho)$ is the cumulative density function of the bivariate standard normal distribution with correlation coefficient ρ.

1.A.3 Put-on-min option

The price of a European put-on-min option with maturity T and strike price K, evaluated at date t, is given by

$$
\begin{aligned}
P_{\min}(S_t^1, S_t^2, K, T-t) &= e^{-r(t-t)}K - C_{\min}(S_t^1, S_t^2, 0, T-t) + \\
&\quad C_{\min}(S_t^1, S_t^2, K, T-t),
\end{aligned}
$$

where $C_{\min}(S_t^1, S_t^2, K, T-t)$ is the price of the European call-on-min option with strike price K and maturity T, evaluated at date t as follows.

$$
\begin{aligned}
C_{\min}(S_t^1, S_t^2, K, T-t) &= S_t^1 e^{-d_1(T-t)}\Phi(d_1^1, d_{12}', -\rho_{12}) + \\
&\quad S_t^2 e^{-d_2(T-t)}\Phi(d_1^2, d_{21}', -\rho_{21}) - \\
&\quad Ke^{-r(T-t)}\Phi(d_2^1, d_2^2, \rho)),
\end{aligned}
$$

where

$$d'_{ij} = \frac{log(S_t^i/S_t^j) + (d_j - d_i - \sigma^2/2)(T-t)}{\sigma\sqrt{(T-t)}}.$$

1.B Transition parameters

The transition parameters $T_{klij}^{\nu\mu}$ for ν and $\mu \in \{0,1\}$, $k \in \{1,\ldots,p\}$, $l \in \{1,\ldots,q\}$, $i \in \{0,\ldots,p\}$, and $j \in \{0,\ldots,q\}$ are calculated as follows.

$$\begin{aligned}
T_{klij}^{00} &= \mathbb{E}^*\left[\mathbb{I}\left((S_{t_{n+1}}^1, S_{t_{n+1}}^2) \in R_{ij}\right) \mid (S_{t_n}^1, S_{t_n}^2) = (a_k, b_l)\right] \\
&= \mathbb{Q}\left[(S_{t_{n+1}}^1, S_{t_{n+1}}^2) \in R_{ij} \mid (S_{t_n}^1, S_{t_n}^2) = (a_k, b_l)\right] \\
&= \int_{x_{k,i}}^{x_{k,i+1}} \int_{y_{l,j}}^{y_{l,j+1}} \phi(z_1, z_2, \rho) dz_1 dz_2 \\
&= \Phi(x_{k,i+1}, y_{l,j+1}, \rho) - \Phi(x_{k,i}, y_{l,j+1}, \rho) - \\
&\quad \Phi(x_{k,i+1}, y_{l,j}, \rho) + \Phi(x_{k,i}, y_{l,j}, \rho),
\end{aligned}$$

where

$$\begin{aligned}
x_{k,i} &= \left(\log\left(a_i/a_k\right) - \left(r - \delta_1 - \sigma_1^2/2\right)\Delta t\right) / \left(\sigma_1\sqrt{\Delta t}\right) \\
y_{l,j} &= \left(\log\left(b_j/b_l\right) - \left(r - \delta_2 - \sigma_2^2/2\right)\Delta t\right) / \left(\sigma_2\sqrt{\Delta t}\right).
\end{aligned}$$

The functions $\phi(\cdot,\cdot,\rho)$ and $\Phi(\cdot,\cdot,\rho)$ are respectively the density and the cumulative density functions of the bivariate standard normal distribution with correlation coefficient ρ. The function $\Phi(\cdot,\cdot,\rho)$ is computed according to Genz (2004).

$$
\begin{aligned}
T^{10}_{klij} &= \mathbb{E}^*\left[S^1_{t_{n+1}}\mathbb{I}\left((S^1_{t_{n+1}},S^2_{t_{n+1}})\in R_{ij}\right) \mid (S^1_{t_n},S^2_{t_n})=(a_k,b_l)\right] \\
&= \int_{x_{k,i}}^{x_{k,i+1}}\int_{y_{l,j}}^{y_{l,j+1}} a_k\exp\left((r-d_1-\sigma_1^2/2)\Delta t+\sigma_1\sqrt{\Delta t}z_1\right)\times \\
&\quad \phi(z_1,z_2,\rho)dz_1dz_2 \\
&= w^1_k\int_{x_{k,i}-\sigma_1\sqrt{\Delta t}}^{x_{k,i+1}-\sigma_1\sqrt{\Delta t}}\int_{y_{l,j}-\rho\sigma_1\sqrt{\Delta t}}^{y_{l,j+1}-\rho\sigma_1\sqrt{\Delta t}}\phi(u_1,u_2,\rho)du_1du_2 \\
&= w^1_k\Big[\Phi(x_{k,i+1}-\sigma_1\sqrt{\Delta t},y_{l,j+1}-\rho\sigma_1\sqrt{\Delta t},\rho)- \\
&\quad \Phi(x_{k,i}-\sigma_1\sqrt{\Delta t},y_{l,j+1}-\rho\sigma_1\sqrt{\Delta t},\rho)- \\
&\quad \Phi(x_{k,i+1}-\sigma_1\sqrt{\Delta t},y_{l,j}-\rho\sigma_1\sqrt{\Delta t},\rho)+ \\
&\quad \Phi(x_{k,i}-\sigma_1\sqrt{\Delta t},y_{l,j}-\rho\sigma_1\sqrt{\Delta t},\rho)\Big],
\end{aligned}
$$

where $w^1_k = a_k\exp\left((r-d_1-\sigma_1^2/2)\Delta t+\sigma_1^2\Delta t/2\right)$.

$$
\begin{aligned}
T^{01}_{klij} &= \mathbb{E}^*\left[S^2_{t_{n+1}}\mathbb{I}\left((S^1_{t_{n+1}},S^2_{t_{n+1}})\in R_{ij}\right) \mid (S^1_{t_n},S^2_{t_n})=(a_k,b_l)\right] \\
&= \int_{x_{k,i}}^{x_{k,i+1}}\int_{y_{l,j}}^{y_{l,j+1}} b_l\exp\left((r-d_2-\sigma_2^2/2)\Delta t+\sigma_2\sqrt{\Delta t}z_2\right)\times \\
&\quad \phi(z_1,z_2,\rho)dz_1dz_2 \\
&= w^2_l\int_{x_{k,i}-\rho\sigma_2\sqrt{\Delta t}}^{x_{k,i+1}-\rho\sigma_2\sqrt{\Delta t}}\int_{y_{l,j}-\sigma_2\sqrt{\Delta t}}^{y_{l,j+1}-\sigma_2\sqrt{\Delta t}}\phi(u_1,u_2,\rho)du_1du_2 \\
&= w^2_l\Big[\Phi(x_{k,i+1}-\rho\sigma_2\sqrt{\Delta t},y_{l,j+1}-\sigma_2\sqrt{\Delta t},\rho)- \\
&\quad \Phi(x_{k,i}-\rho\sigma_2\sqrt{\Delta t},y_{l,j+1}-\sigma_2\sqrt{\Delta t},\rho)- \\
&\quad \Phi(x_{k,i+1}-\rho\sigma_2\sqrt{\Delta t},y_{l,j}-\sigma_2\sqrt{\Delta t},\rho)+ \\
&\quad \Phi(x_{k,i}-\rho\sigma_2\sqrt{\Delta t},y_{l,j}-\sigma_2\sqrt{\Delta t},\rho)\Big],
\end{aligned}
$$

where $w_l^2 = b_l \exp\Big((r - d_2 - \sigma_2^2/2)\Delta t + \sigma_2^2 \Delta t/2\Big)$.

$$
\begin{aligned}
T^{11}_{klij} &= \mathbb{E}^*\Big[S^1_{t_{n+1}} S^2_{t_{n+1}} \mathbb{I}\Big((S^1_{t_{n+1}}, S^2_{t_{n+1}}) \in R_{ij}\Big) \mid (S^1_{t_n}, S^2_{t_n}) = (a_k, b_l)\Big] \\
&= \int_{x_{k,i}}^{x_{k,i+1}} \int_{y_{l,j}}^{y_{l,j+1}} a_k \exp\Big((r - d_1 - \sigma_1^2/2)\Delta t + \sigma_1 \sqrt{\Delta t} z_1\Big) \times \\
&\quad b_l \exp\Big((r - d_2 - \sigma_2^2/2)\Delta t + \sigma_2 \sqrt{\Delta t} z_2\Big) \phi(z_1, z_2, \rho) dz_1 dz_2 \\
&= w_k^1 w_l^2 \exp(\rho \sigma_1 \sigma_2 \Delta t) \times \\
&\quad \int_{x_{k,i}-(\sigma_1+\rho\sigma_2)\sqrt{\Delta t}}^{x_{k,i+1}-(\sigma_1+\rho\sigma_2)\sqrt{\Delta t}} \int_{y_{l,j}-(\rho\sigma_1+\sigma_2)\sqrt{\Delta t}}^{y_{l,j+1}-(\rho\sigma_1+\sigma_2)\sqrt{\Delta t}} \phi(u_1, u_2, \rho) du_1 du_2 \\
&= w_k^1 w_l^2 \exp(\rho \sigma_1 \sigma_2 \Delta t) \times \\
&\quad \Big[\Phi(x_{k,i+1} - (\sigma_1 + \rho\sigma_2)\sqrt{\Delta t}, y_{l,j+1} - (\rho\sigma_1 + \sigma_2)\sqrt{\Delta t}, \rho) - \\
&\quad \Phi(x_{k,i} - (\sigma_1 + \rho\sigma_2)\sqrt{\Delta t}, y_{l,j+1} - (\rho\sigma_1 + \sigma_2)\sqrt{\Delta t}, \rho) - \\
&\quad \Phi(x_{k,i+1} - (\sigma_1 + \rho\sigma_2)\sqrt{\Delta t}, y_{l,j} - (\rho\sigma_1 + \sigma_2)\sqrt{\Delta t}, \rho) + \\
&\quad \Phi(x_{k,i} - (\sigma_1 + \rho\sigma_2)\sqrt{\Delta t}, y_{l,j} - (\rho\sigma_1 + \sigma_2)\sqrt{\Delta t}, \rho)\Big].
\end{aligned}
$$

BIBLIOGRAPHY

Andersen, L. and Broadie, M. (2004). Primal-dual simulation algorithm for pricing multidimensional American options. *Management Science*, 50(9):1222–1234.

Bally, V. and Pages, G. (2003a). Error analysis of the optimal quantization algorithm for obstacle problems. *Stochastic Processes and their Applications*, 106(1):1–40.

Bally, V. and Pages, G. (2003b). A quantization algorithm for solving multidimensional discrete-time optimal stopping problems. *Bernoulli*, 9(6):1003–1049.

Bally, V. and Printems, J. (2005). A quantization tree method for pricing and hedging multidimensional American options. *Mathematical Finance*, 15(1):119–168.

Barraquand, J. and Martineau, D. (1995). Numerical valuation of high dimensional multivariate American securities. *Journal of Financial and Quantitative Analysis*, 30(03):383–405.

Ben-Ameur, H., Breton, M., and L'Ecuyer, P. (2002). A dynamic programming procedure for pricing American-style asian options. *Management Science*, 48(5):625–643.

Ben-Ameur, H., Chérif, R., and Rémillard, B. (2016). American-style options in jump-diffusion models: estimation and evaluation. *Quantitative Finance. In Press.*

Berridge, S. and Schumacher, J. (2008). An irregular grid approach for pricing high-dimensional american options. *Journal of Computational and Applied Mathematics*, 222(1):94–111.

Boyle, P., Broadie, M., and Glasserman, P. (1997). Monte Carlo methods for security pricing. *Journal of Economic Dynamics and Control*, 21(8):1267–1321.

Boyle, P. P. (1988). A lattice framework for option pricing with two state variables. *Journal of Financial and Quantitative Analysis*, 23(1):1–12.

Boyle, P. P., Evnine, J., and Gibbs, S. (1989). Numerical evaluation of multivariate contingent claims. *Review of Financial Studies*, 2(2):241–250.

Broadie, M. and Glasserman, P. (1997). Pricing American-style securities using simulation. *Journal of Economic Dynamics and Control*, 21(8):1323–1352.

Carriere, J. F. (1996). Valuation of the early-exercise price for options using simulations and nonparametric regression. *Insurance: Mathematics and Economics*, 19(1):19–30.

Del Moral, P., Rémillard, B., and Rubenthaler, S. (2012). Monte Carlo approximations of American options that preserve monotonicity and convexity. In *Numerical Methods in Finance*, pages 115–143. Springer.

Detemple, J., Feng, S., and Tian, W. (2003). The valuation of American call options on the minimum of two dividend-paying assets. *Annals of Applied Probability*, 13(3):953–983.

Dockendorf, J. and Paxson, D. A. (2015). Sequential real rainbow options. *The European Journal of Finance*, 21(10-11):867–892.

Genz, A. (2004). Numerical computation of rectangular bivariate and trivariate normal and t probabilities. *Statistics and Computing*, 14(3):251–260.

Hartley, P. (2000). Pricing a multi-asset American option. Working paper. University of Bath.

Haugh, M. B. and Kogan, L. (2004). Pricing American options: a duality approach. *Operations Research*, 52(2):258–270.

Jin, X., Li, X., Tan, H. H., and Wu, Z. (2013). A computationally efficient state-space partitioning approach to pricing high-dimensional American options via dimension reduction. *European Journal of Operational Research*, 231(2):362–370.

Jin, X., Tan, H. H., and Sun, J. (2007). A state-space partitioning method for pricing high-dimensional American-style options. *Mathematical Finance*, 17(3):399–426.

Johnson, H. (1987). Options on the maximum or the minimum of several assets. *Journal of Financial and Quantitative Analysis*, 22(3):277–283.

Kamrad, B. and Ritchken, P. (1991). Multinomial approximating models for options with k state variables. *Management Science*, 37(12):1640–1652.

Longstaff, F. A. and Schwartz, E. S. (2001). Valuing American options by simulation: a simple least-squares approach. *The Review of Financial Studies*, 14(1):113–147.

Raymar, S. B. and Zwecher, M. J. (1997). A Monte Carlo valuation of American call options on the maximum of several stocks. *Journal of Derivatives*, 5(1):7–23.

Rogers, L. C. G. (2002). Monte Carlo valuation of American options. *Mathematical Finance*, 12(3):271–286.

Stulz, R. (1982). Options on the minimum or the maximum of two risky assets: analysis and applications. *Journal of Financial Economics*, 10(2):161–185.

Tilley, J. A. (1993). Valuing American options in a path simulation model. *Transactions of the Society of Actuaries*, 45:83–104.

Tsitsiklis, J. N. and Van Roy, B. (1999). Optimal stopping of Markov processes: Hilbert space theory, approximation algorithms, and an application to pricing high-dimensional financial derivatives. *IEEE Transactions on Automatic Control,*, 44(10):1840–1851.

CHAPTER 2

A TWO-FACTOR STRUCTURAL MODEL FOR VALUING CORPORATE SECURITIES

Malek Ben-Abdellatif,[1] Hatem Ben Ameur,[2] Bruno Rémillard[3]

Abstract

We develop a general structural model for valuing risky corporate debts that takes into account both default and interest rate risk. We propose a two-dimensional model in which the state variables are the value of the firm's assets and the short-term interest rate. The former follows a lognormal process and the latter a mean-reverting Gaussian process. Our methodology is based on dynamic programming and finite elements. We use parallel computing to enhance its efficiency. Our model accommodates flexible debt structure, multiple seniority classes, tax benefits, and bankruptcy costs. The results we obtain are consistent with empirical evidence documented in the literature.

Keywords: Credit risk; Structural model; stochastic interest rate; dynamic programming; finite elements; parallel computing.

2.1 Introduction

We propose a structural model for valuing risky debts when the interest rate is stochastic. Our methodology is based on two-dimensional dynamic programming coupled with finite elements. We use parallel computing to enhance our procedure's

---

1. Malek Ben-Abdellatif is a Ph.D. student at HEC Montréal.

2. Hatem Ben-Ameur is a Professor at HEC Montréal in the Department of Decision Sciences, and member of the GERAD.

3. Bruno Rémillard is a Professor at HEC Montréal in the Department of Decision Sciences, and member of the CRM and GERAD.

Partial funding in support of this work was provided by the Natural Sciences and Engineering Research Council of Canada, the Fonds pour la formation de chercheurs et l'aide à la recherche du Gouvernement du Québec, and the Institut de Finance Mathématique du Québec.

efficiency. Classical structural models assume a fixed interest rate, but this assumption is too simplistic as interest rates are stochastic in practice, particularly since we observe long maturities for corporate debts. Empirical evidence suggests that the interest rate and credit risk are negatively correlated (Litterman and Scheinkman, 1991, Collin-Dufresne et al., 2001). The activity of the underlying company and its capital structure policy can be sensitive to the term structure of interest rates. The empirical work of Longstaff and Schwartz (1995) shows that bonds with similar credit ratings, but in different industries, have widely varying credit spreads. Theses differences are explained by the manifold correlations with interest rates. Contrary to the traditional approach, which implies that credit spreads depend only on an asset-value factor, Longstaff and Schwartz (1995) show that credit spreads for corporate bonds are driven by an asset-value factor and an interest-rate factor; the dependence between the two factors plays a crucial role in determining credit spreads. It is thus important to include interest rate uncertainty in the credit risk modeling framework.

Structural models are based on the pioneer work of Merton (1974) who considered the firm's assets to follow geometric-Brownian motion. Default occurs if the firm's assets are insufficient to pay the debt at maturity. Considering the debt to be a pure bond, he uses option-pricing theory; the firm's equity is evaluated as a European call option on the firm's assets, with the same maturity as the bond, and a strike price equal to the principal amount. Although simple and unrealistic, this work has generated several developments in credit risk modeling and is the basis for more general models.

Black and Cox (1976) propose a barrier-triggered default which allows for default to happen before maturity of the debt. Several authors consider more complex debt structures or include frictions (Ericsson and Reneby, 1998, Collin-Dufresne and Goldstein, 2001, Hsu et al., 2010, Geske, 1977), endogenous default barriers (Leland, 1994, Anderson and Sundaresan, 1996, Leland and Toft, 1996, Mella-Barral and Perraudin, 1997, François and Morellec, 2004), and jumps in the firm's asset process (Zhou, 2001, Chen and Kou, 2009).

To incorporate interest rate risk in the corporate debt valuation, various articles include a stochastic interest rate in structural models. Shimko et al. (1993) add a stochastic short-term interest rate that evolves according to Vasicek's (1977) model to Merton's (1974) model. In this case, a closed-form solution is available as the problem becomes equivalent to pricing an European call option on a stock under the stochastic interest rate.

Kim et al. (1993) and Longstaff and Schwartz (1995) extend the Black and Cox (1976) model. The former considers a CIR dynamic following Cox et al. (1985) for the short-term interest rate while the latter uses Vasicek's (1977) model; both derive a quasi-closed form for the debt value. Cathcart and El-Jahel (1998) and Briys and De Varenne (1997) propose some corrections to the Longstaff and Schwartz's (1995) model. The first adds a CIR process for the short-term rate to avoid having positive probability that the interest rate becomes negative. The second corrects for weaknesses such as bondholders recovering an amount that, in case of default, does not depend on the remaining firm's asset value. All these models consider very simple settings regarding the firm's capital structure and the default mechanism. Allowing endogenous default barriers or more general debt structures requires using a numerical approach to solve the problem.

We extend Altieri and Vargiolu (2001) and Ayadi et al. (2016) by adding a stochastic interest rate to a general structural model which allows for a flexible debt structure with multiple seniority classes, and accounts for bankruptcy costs and tax benefits. We use a mean-reverting Gaussian process for the short-term interest rate as proposed by Vasicek (1977). The proposed methodology is based on a two-dimensional dynamic program coupled with finite elements. As this procedure is time demanding, we use parallel computing to expedite our procedure and improve its efficiency. Our results demonstrate convergence and remain consistent with empirical evidence documented in the literature.

This paper is organized as follows: Section 2.2 presents our model, Section 2.3 describes our dynamic program, Section 2.4 shows our numerical investigation, and Section 2.5 concludes our paper.

2.2 Model and notations

We propose a structural model for valuing risky debt by allowing for both default risk and interest rate risk. The stochastic short-term interest rate r_t evolves according to a mean-reverting Gaussian process as in Vasicek's (1977) model

$$dr_t = \alpha(\beta - r_t)dt + \sigma_r dZ_t^1, \tag{2.1}$$

where β is the long-term mean level, α is the speed of reversion to this level, and σ_r is the instantaneous volatility. The firm's assets value V_t moves according to geometric-Brownian motion

$$\frac{dV_t}{V_t} = (r_t - \delta)dt + \sigma_V(\rho dZ_t^1 + \sqrt{1-\rho^2}dZ_t^2), \tag{2.2}$$

where δ is the firm's payout rate and σ_v is its assets volatility. Both dynamics are under the risk neutral measure $\mathbb{Q}$. Z_t^1 and Z_t^2 are two independent Brownian motions and ρ represents the correlation between the two processes.

Consider that the firm's capital structure contains a portfolio of senior and junior bonds and a common stock. The firm makes coupon payments to the bondholders which results in collecting tax benefits. The firm also pays bankruptcy costs in case of default. The model assumes that the stockholders determine the time of default by maximizing the firm's total value subject to the limited liability constraint. Let $\mathscr{P} = \{t_0, t_1, \ldots, t_n, \ldots, t_N\}$ be a set of payment dates, and let $(\Omega, \mathscr{F}_t, \mathbb{P})$ be a complete probability space. For each $n \in \{0, \ldots, N\}$, $k \in \{0, \ldots, N+1\}$, where k is the bankruptcy time, set $r_n^k = -\int_{t_n}^{t_k} r_s ds$; the discount factor is then $e^{-r_n^k}$. The case $k = N+1$ means that the firm survives until date t_N. The value functions, in terms of the bankruptcy time k, are expressed as follows:

Bankruptcy costs: The costs connected to default are equal to wV_τ, where

$w \in (0,1)$ is a fixed fraction. The value of bankruptcy costs at time t_n is given by

$$BC_k^{(n)} = \begin{cases} 0, & k < n \text{ or } k = N+1, \\ we^{-r_n^k} V_k, & n \le k \le N. \end{cases}$$

Debt: At each date t_n, the firm is committed to pay $d_n^{(sen)} + d_n^{(jun)} = d_n$ to its creditors, where $d_n^{(sen)}$ and $d_n^{(jun)}$ are the payments due to the senior and junior bondholders, respectively. These payments include interest as well as principal payments. The interest payment is denoted d_n^{int}. The last payment dates of the senior and junior debts are indicated by T^s and T^j, with $0 \le T^s \le T^j = T$. The senior and junior debts are

$$DS_k^{(n)} = \begin{cases} 0, & k < n, \\ e^{-r_n^k} \min\left\{(1-w)V_k, d_k^{(sen)}\right\} + \sum_{j=n}^{k-1} e^{-r_n^j} d_j^{(sen)}, & n \le k \le N, \\ \sum_{j=n}^{N} e^{-r_n^j} d_j^{(sen)}, & k = N+1, \end{cases}$$

and

$$DJ_k^{(n)} = \begin{cases} 0, & k < n, \\ r_n^k \max\left\{(1-w)V_k - d_k^{(sen)}, 0\right\} + \sum_{j=n}^{k-1} e^{-r_n^j} d_j^{(jun)}, & n \le k \le N, \\ \sum_{j=n}^{N} e^{-r_n^j} d_j^{(jun)}, & k = N+1. \end{cases}$$

where $\sum_{j=n}^{n-1} = 0$, by convention.

The total debt at time t_n is then

$$D_k^{(n)} = \begin{cases} 0, & k < n, \\ (1-w)e^{-r_n^k} V_k + \sum_{j=n}^{k-1} e^{-r_n^j} d_j, & n \le k \le N, \\ \sum_{j=n}^{N} e^{-r_n^j} d_j, & k = N+1. \end{cases}$$

Tax benefits: The tax benefits associated with the cost of debt are proportional to the interest payment d_n^{int}. Let $r_n^c \in [0,1]$ be the periodic corporate tax rate over $[t_n, t_{n+1}]$ and $tb_n = r_n^c d_n^{int}$. The tax benefits are then

$$TB_k^{(n)} = \begin{cases} 0, & k < n, \\ \sum_{j=n}^{k-1} e^{-r_n^k} tb_j, & n \leq k \leq N+1. \end{cases}$$

The total value of the firm: The total value of the firm represents the assets' value increased by the tax benefits, net of the bankruptcy costs,

$$\begin{aligned} W_k^{(n)} &= V_n + TB_k^{(n)} - BC_k^{(n)} \\ &= \begin{cases} 0, & k < n, \\ V_n + \sum_{j=n}^{k-1} e^{-r_n^j} tb_j - we^{-r_n^k} V_k, & n \leq k \leq N, \\ V_n + \sum_{j=n}^{N} e^{-r_n^j} tb_j, & k = N+1. \end{cases} \end{aligned}$$

Equity value: In case of survival at date t_n, the stockholders receive the total value of the firm minus the total debt value

$$\mathscr{E}_k^{(n)} = W_k^{(n)} - D_k^{(n)}.$$

Let $\mathscr{T}$ be the set of stopping times with values in $\{0, \ldots, N+1\}$. As a result, for any stopping time $\tau \in \mathscr{T}$ with $\tau \geq n$, one obtains

$$E\left(\mathscr{E}_\tau^{(n)} | \mathscr{F}_n\right) = \mathscr{B}_n^{(\tau)} \mathbf{1}(\tau > n),$$

where $\mathscr{B}_N^{(\tau \vee N)} = \mathscr{B}_N = V_N + tb_N - d_N$ and

$$\mathscr{B}_n^{(\tau)} = V_n + tb_n - d_n - E\left(e^{-r_n^{n+1}} V_{n+1} | \mathscr{F}_n\right) + E\left(e^{-r_n^{n+1}} \mathscr{E}_{\tau \vee (n+1)}^{(n+1)} | \mathscr{F}_n\right),$$

for all $n \in \{0, \ldots, N-1\}$.

Definition 1.

$$\mathscr{T}_n = \left\{ \tau \in \mathscr{T}; \tau \geq n, \; \{\tau > k\} \subset \left\{ E\left(\mathscr{E}^{(k)}_{\tau \vee k} | \mathscr{F}_k \right) > 0 \right\}, \;\; for \; k \geq n \right\}.$$

Finally, define $J^{(n)}_\tau = TB^{(n)}_\tau - BC^{(n)}_\tau$, and set

$$\bar{J}_n = \sup_{\tau \in \mathscr{T}_n} E\left(J^{(n)}_\tau \middle| \mathscr{F}_n \right),$$

for all $n \in \{0, \ldots, N\}$. Note that $\sup_{\tau \in \mathscr{T}_n} E\left\{ W^{(n)}_\tau \middle| \mathscr{F}_n \right\} = V_n + \bar{J}_n$.

The main aim is to find a sequence of stopping times $\tau^\star_n \in \mathscr{T}_n$, corresponding to optimal bankruptcy times, so that the total expected wealth at time n is maximized, that is $V_n + \bar{J}_n = E\left\{ W^{(n)}_{\tau^\star_n} | \mathscr{F}_n \right\}$. The solution is provided by Ben-Abdellatif et al. (2016) in the following theorem.

Theorem 1. *Set* $\mathscr{E}_N = \max(V_N + tb_N - d_N, 0)$*. For any* $k \in \{0, \ldots, N-1\}$*, set*

$$\mathscr{E}_k = \max\left\{ V_k + tb_k - d_k - E\left(e^{-r^{k+1}_k} V_{k+1} | \mathscr{F}_k \right) + E\left(e^{-r^{k+1}_k} \mathscr{E}_{k+1} | \mathscr{F}_k \right), 0 \right\}.$$

Next, define

$$\tau^\star_k = \begin{cases} N+1, & \text{if } \mathscr{E}_j > 0 \text{ for all } j \in \{k, \ldots, N\}, \\ \min\{k \leq j \leq N; \mathscr{E}_j = 0\}, & \text{otherwise.} \end{cases}$$

Then

$$\bar{J}_N = E\left(J^{(N)}_{\tau^\star_N} | \mathscr{F}_N \right) = -\alpha V_N \mathbf{1}(\mathscr{E}_N = 0) + b_N \mathbf{1}(\mathscr{E}_N > 0),$$

and for all $k \in \{0, \ldots, N-1\}$*,*

$$\begin{aligned} \bar{J}_k &= E\left(J^{(k)}_{\tau^\star_k} | \mathscr{F}_k \right) \\ &= -\alpha V_k \mathbf{1}(\mathscr{E}_k = 0) + \left\{ tb_k + E\left(e^{-r^{k+1}_k} \bar{J}_{k+1} | \mathscr{F}_k \right) \right\} \mathbf{1}(\mathscr{E}_k > 0). \end{aligned}$$

The proof of Theorem 1 is given in Ben-Abdellatif et al. (2016). Now suppose that $V_n = V(t_n)$. Further, set $\mathscr{F}_n = \sigma\{r(u), V(u); 0 \leq u \leq t_n\}$.

Note that our model satisfies the **Markovian hypothesis**, meaning that there is an expectation operator T_n so that for any integrable function Ψ on $\mathbb{R} \times [0, \infty)$,

$$E\left[e^{-\int_{t_n}^{t_{n+1}} r(u)du} \Psi\{r(t_{n+1}), V(t_{n+1})\} | \mathscr{F}_n\right] = T_n \Psi\{r(t_{n+1}), V(t_n + 1)\}. \tag{2.3}$$

In our setting, this expectation operator is calculated as follows

$$T_n \Psi\{r(t_{n+1}), V(t_{n+1})\} = B(t_n, t_{n+1}) E^* \left[\Psi\{r(t_{n+1}), V(t_{n+1})\} | \mathscr{F}_n\right],$$

where E^* is the expectation under the forward measure, and $B(t_n, t_{n+1})$ is the price of a zero-coupon bond with maturity t_{n+1} at time t_n. The change of measure using the forward measure is done according to Jamshidian (1989) and is described in Appendix 2.A. For this setting, the following proposition from Ben-Abdellatif et al. (2016) gives the expression of the value functions.

Proposition 1. *Set $r = r(t_n)$ and $v = V(t_n)$. Under the Markovian hypothesis, for $k = N$, one has*

$$\begin{aligned}
\mathscr{E}_N(r, v) &= \max(v + tb_N - d_N, 0), & (2.4)\\
D_N(r, v) &= (1 - w)v \mathbf{1}\{\mathscr{E}_N(r, v) = 0\} + d_N \mathbf{1}\{\mathscr{E}_N(r, v) > 0\}, &\\
DS_N(r, v) &= \min\left\{(1 - w)v, d_N^{(sen)}\right\} \mathbf{1}\{\mathscr{E}_N(r, v) = 0\} + &\\
&\quad d_N^{(sen)} \mathbf{1}\{\mathscr{E}_N(r, v) > 0\}, & (2.5)\\
DJ_N(r, v) &= \max\left\{(1 - w)v - d_N^{(sen)}, 0\right\} \mathbf{1}\{\mathscr{E}_N(r, v) = 0\} + &\\
&\quad d_N^{(jun)} \mathbf{1}\{\mathscr{E}_N(r, v) > 0\}, & (2.6)\\
TB_N(r, v) &= tb_N \mathbf{1}\{\mathscr{E}_N(r, v) > 0\}, & (2.7)\\
BC_N(r, v) &= wv \mathbf{1}\{\mathscr{E}_N(r, v) = 0\}, & (2.8)
\end{aligned}$$

and for any $k \in \{0,\ldots,N-1\}$

$$\begin{aligned}
\mathscr{E}_k(r,v) &= \max\left\{b_k - d_k + T_k\mathscr{E}_{k+1}(r,v), 0\right\}, && (2.9)\\
D_k(r,v) &= (1-w)v\mathbf{1}\{\mathscr{E}_k(r,v)=0\} + \\
&\quad \left\{d_k + T_k D_{k+1}(r,v)\right\}\mathbf{1}\{\mathscr{E}_k(r,v)>0\}, \\
DS_k(r,v) &= \min\left\{(1-w)v, d_k^{(sen)}\right\}\mathbf{1}\{\mathscr{E}_k(r,v)=0\} + \\
&\quad \left\{d_k^{(sen)} + T_k DS_{k+1}(r,v)\right\}\mathbf{1}\{\mathscr{E}_k(r,v)>0\}, && (2.10)\\
DJ_k(r,v) &= \max\left\{(1-w)v - d_k^{(sen)}, 0\right\}\mathbf{1}\{\mathscr{E}_k(r,v)=0\} + \\
&\quad \left\{d_k^{(jun)} + T_k DJ_{k+1}(r,v)\right\}\mathbf{1}\{\mathscr{E}_k(r,v)>0\}, && (2.11)\\
TB_k(r,v) &= \{tb_n + T_k TB_{k+1}\}\mathbf{1}\{\mathscr{E}_k(r,v)>0\}, && (2.12)\\
BC_k(r,v) &= \alpha V_n\mathbf{1}\{\mathscr{E}_k(r,v)=0\} + T_k BC_{k+1}\mathbf{1}\{\mathscr{E}_k(r,v)>0\}. && (2.13)
\end{aligned}$$

2.3 Dynamic programming

The implementation of the optimal stopping time problem presented in Section 2.2 is done by using dynamic programming coupled with finite elements and bilinear interpolations. Parallel computing is used to accelerate the execution time of our program and enhance its efficiency.

Let $\mathscr{G}$ be a set of grid points $\{(a_1,b_1),(a_1,b_2),\ldots,(a_p,b_q)\}$ such that $\max(\Delta a_k,\Delta b_l) \to 0$ and $\mathbb{Q}[(V_t,r_t) \in [a_p,\infty)\times\mathbb{R}_+^* \cup \mathbb{R}_+^* \times [b_q,\infty)] \to 0$, when p and $q\to\infty$. Let $a_0 = b_0 = 0$ and $a_{p+1} = b_{q+1} = \infty$. The rectangle $[a_i,a_{i+1})\times[b_j,b_{j+1})$ is designated by R_{ij}.

Dynamic programming acts as follows.

1. At date $t_N = T$, the value functions are known in closed form and are computed following Eq. (2.4), (2.6), (2.5), (2.7) and (2.8).
2. At each date t_n, suppose that an approximation of each value function is available at a future decision date t_{n+1} on $\mathscr{G}$, indicated by $\widetilde{\Psi}_{n+1}(a_k,b_l)$, for $k=1,\ldots,p$ and $l=1,\ldots,q$, where Ψ_n represents TB_n, BC_n, DS_n, DJ_n, or $\mathscr{E}_n$. Use a bilinear piecewise polynomial, and interpolate each value function $\widetilde{\Psi}_{n+1}$

from $\mathscr{G}$ to the overall state space $[0,\infty)^2$ by setting

$$\widehat{\Psi}_{n+1}(x,e^y) = \sum_{i=0}^{p}\sum_{j=0}^{q}\left(\alpha_{ij}^{n+1} + \beta_{ij}^{n+1}x + \gamma_{ij}^{n+1}e^y + \delta_{ij}^{n+1}xe^y\right)\mathbb{I}\left((x,y)\in R_{ij}\right).$$

The local coefficients of each value function f_{n+1}, α_{ij}^{n+1}, β_{ij}^{n+1}, γ_{ij}^{n+1}, and δ_{ij}^{n+1}, for $i=0,\ldots,p$ and $j=0,\ldots,q$, are those of the bilinear interpolation.

3. Approximate every expected discounted value function at t_n on $\mathscr{G}$

$$\begin{aligned}
&E\left[e^{-\int_{t_n}^{t_{n+1}} r_s ds}\widehat{\Psi}_{n+1}(V_{t_{n+1}}, r_{t_{n+1}}) \mid (V_{t_n}, r_{t_n}) = (a_k, b_l)\right] \\
&= B(t_n, t_{n+1})E^*\left[\widehat{\Psi}_{n+1}(V_{t_{n+1}}, r_{t_{n+1}}) \mid (V_{t_n}, r_{t_n}) = (a_k, b_l)\right] \\
&= B(t_n, t_{n+1})\sum_{i,j}\left(\alpha_{ij}^{n+1}T_{klij}^{00} + \beta_{ij}^{n+1}T_{klij}^{10} + \gamma_{ij}^{n+1}T_{klij}^{01} + \delta_{ij}^{n+1}T_{klij}^{11}\right),
\end{aligned} \tag{2.14}$$

where the transition tables T^{00}, T^{10}, T^{01}, and T^{11} are defined by

$$\begin{aligned}
T_{klij}^{\nu\mu} &= \mathbb{E}^*\Big[(V_{t_{n+1}})^{\nu}(e^{r_{t_{n+1}}})^{\mu}\mathbb{I}\left((V_{t_{n+1}}, r_{t_{n+1}})\in R_{ij}\right) \mid \\
&\qquad (V_{t_n}, r_{t_n}) = (a_k, b_l)\Big], \quad \text{for } \nu \text{ and } \mu \in \{0,1\}.
\end{aligned}$$

For example, T_{klij}^{00} represents the transition probability that the Markov process (V,r) moves from (a_k, b_l) at t_n and visits the rectangle R_{ij} at t_{n+1}. Closed-form solutions for the transition parameters are given in Appendix 3.2.

4. Compute the value functions at t_n on $\mathscr{G}$ following Eq. (2.9), Eq. (2.11), Eq. (2.10), Eq. (2.12) and Eq. (2.13), using Eq. (2.14).

5. Go to step 2 and repeat until $n=0$.

2.4 Numerical investigation

Parallel computing uses multiple central processing units (CPUs) simultaneously to accelerate complex computations. The Message Passing Interface (MPI) library allows the computing process to exchange information between the running CPUs in order to achieve a given job. We parallelize our dynamic program by submitting the computation tasks associated to a given number of grid points to each available CPU. The algorithm used to parallelize our dynamic program is described in detail in appendix 3.B. This approach allows us to drastically reduce computation times to a reasonable level. Our numerical investigation presented in this section are based on a grid size of 300^2. A price calculation takes in average two minutes using parallel computing.

We use the supercomputer Briarée managed by Calcul Québec and Compute Canada [4]. The code lines are written in C and compiled with GCC. We use the MPI library to access parallel computing.

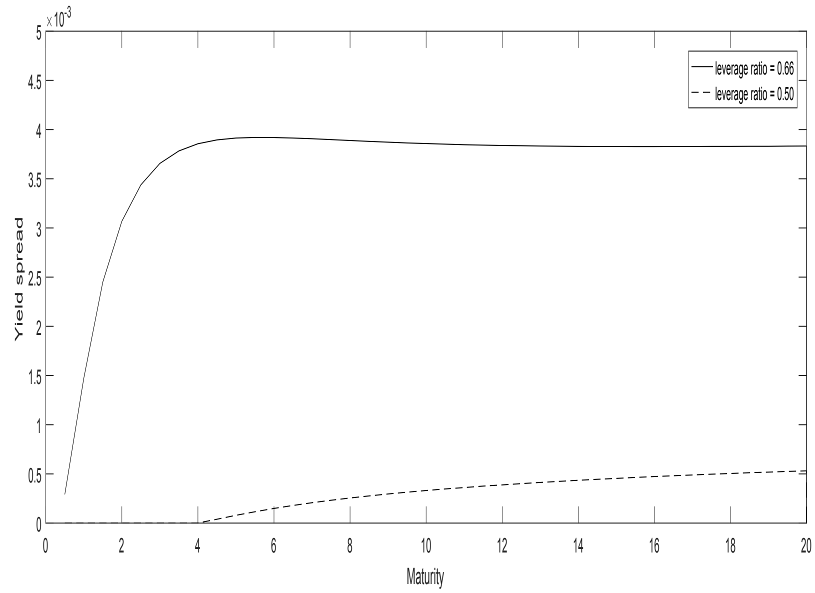

Figure 2.1: Credit spreads for an 8% bond for different leverage ratios. The parameters used are $r = 0.04, \alpha = 1, \beta = 0.06, \sigma_r = 0.03, \rho = -0.25, \sigma_v = 0.2, w = 0$, and $r^c = 0$.

We consider similar parameters to those in Longstaff and Schwartz (1995) for the interest rate dynamic as plausible parameter values. Figure 2.1 presents the term structure of credit spreads when the firm's leverage ratio (debt principle/firm's

4. The operation of this supercomputer is funded by the Canada Foundation for Innovation (CFI), Ministère de l'Économie, de la Science et de l'Innovation du Québec (MESI) and the Fonds de recherche du Québec - Nature et technologies (FRQ-NT).

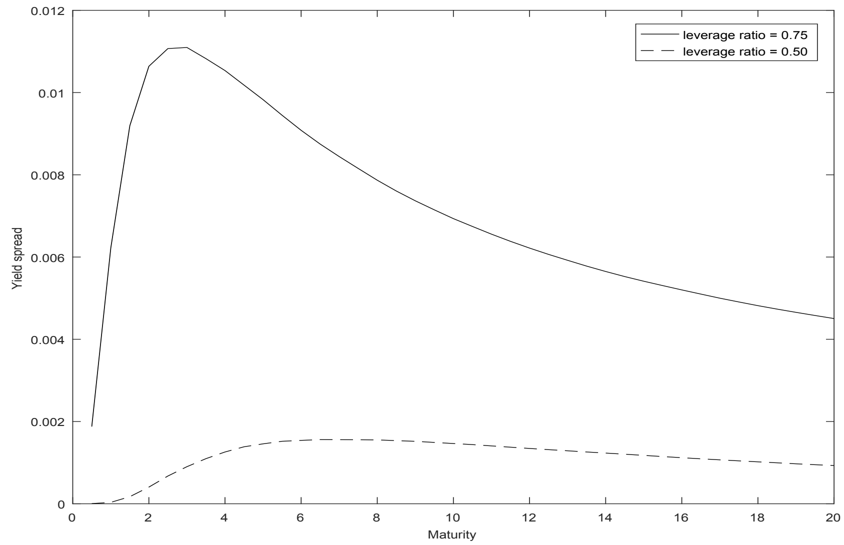

Figure 2.2: Credit spreads for an 8% bond for different leverage ratios. The parameters used are $r = 0.04, \alpha = 1, \beta = 0.06, \sigma_r = 0.03, \rho = -0.25, \sigma_v = 0.2, w = 0.3$, and $r^c = 0.35$.

assets) is changed, but without tax benefits and bankruptcy costs. Credit spreads are greater for a higher leverage ratio, which corresponds to more risky debt. We also observe that credit spreads increase with maturity. Figure 2.2 considers the case with bankruptcy costs and corporate taxes. The term structure of credit spreads is monotone increasing for firms with a low leverage ratio associated with good rated bonds. Conversely, the credit spreads' term structure is hump shaped for firms with higher leverage ratios, thus corresponding to bonds with low ratings. This is consistent with empirical evidence, as explained by Sarig and Warga (1989) and Kim et al. (1993).

Figure 2.3 plots the term structure of the credit spread for various levels of the current interest rate r, and shows a negative relation between credit spreads and the level of the short-term interest rate. An increase in r tends to reduce the default probability as it affects the drift on the firm's assets dynamic, reducing the yield spread. However, the magnitude of decrease in the credit spread depends on the correlation between asset returns and changes in the interest rate. As shown in Figure 2.4, the credit spread increases when the correlation increases. As

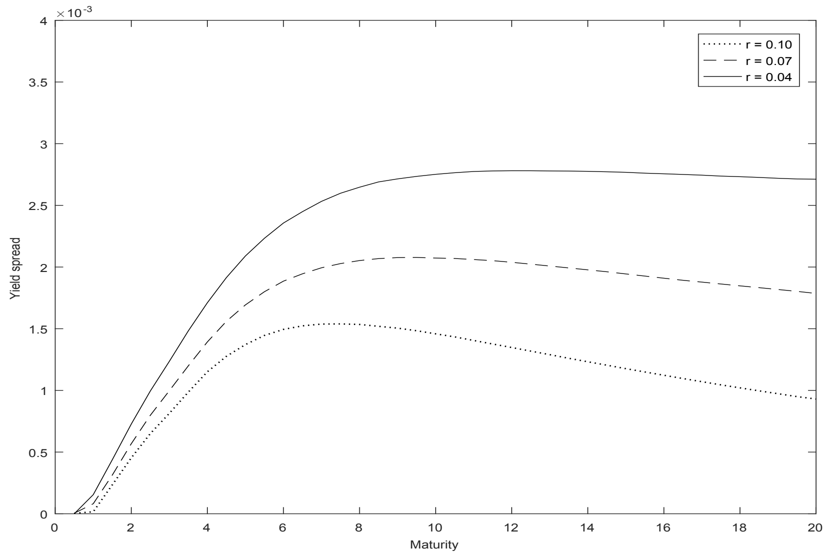

Figure 2.3: Credit spreads for an 8% bond for different values of r. The parameters used are $\alpha = 1, \beta = 0.06, \sigma_r = 0.03, \rho = -0.25, \sigma_v = 0.2, w = 0.3, r^c = 0.35$, and leverage ratio $= 0.5$.

explained by Longstaff and Schwartz (1995), differences in the duration of bonds across industries is related to the differences in correlation with the interest rate level.

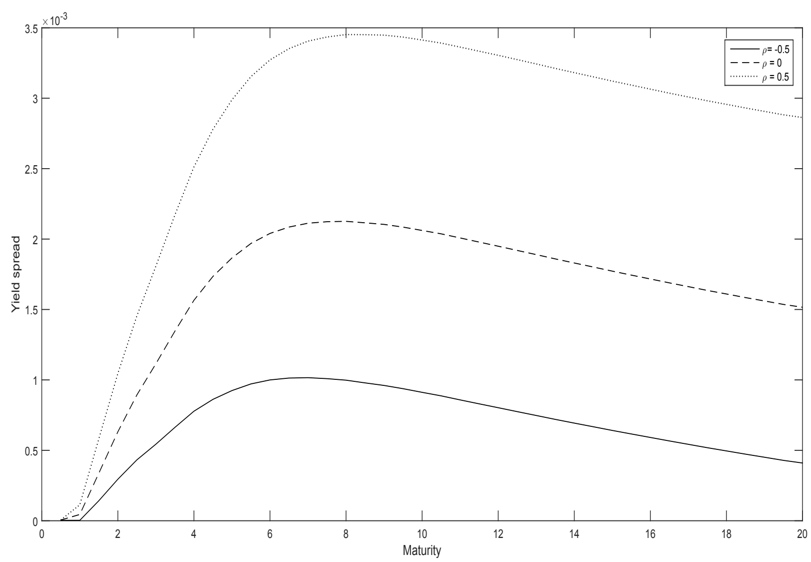

Figure 2.4: Credit spreads for an 8% bond for different values of ρ. The parameters used are $r = 0.04, \alpha = 1, \beta = 0.06, \sigma_r = 0.03, \sigma_v = 0.2, w = 0.3, r^c = 0.35$, and leverage ratio $= 0.5$.

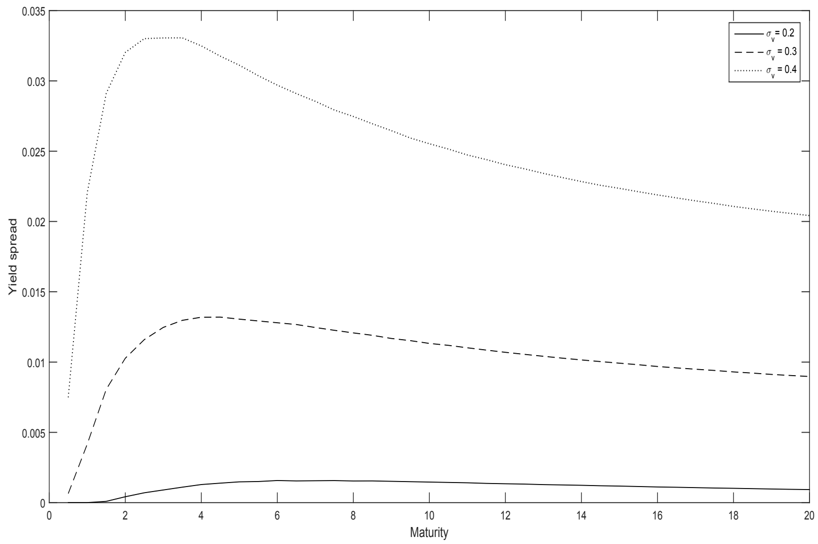

Figure 2.5: Credit spreads for an 8% bond for different values of σ_v. The parameters used are $r = 0.04, \alpha = 1, \beta = 0.06, \sigma_r = 0.03, \rho = -0.25, w = 0.3, r^c = 0.35$, and leverage ratio $= 0.5$.

Figure 2.5 plots the credit spread for different values of volatility for the firm's assets σ_r. As the latter increases, the credit spread increases. The term structure of credit spreads is monotone increasing for firms with low risk activities, while it is hump shaped for more risky firms.

Our paper does not address the estimation problem but it is interesting to notice that under the structural credit model, it remains an issue. The main difficulty of the estimation problem is that the firm's assets value cannot be directly observed. This is further complicated by the fact that the data samples only comprise of surviving firms. Several approaches were proposed to tackle the estimation problem. We briefly discuss the main two methodologies; the first one is based on a transformed-data-maximum likelihood method. Duan (1994, 2000) was the pioneer and proposes a likelihood function based on the observed equity prices. He views them as a sample of transformed data using the equity pricing equation. Later, and under the same spirit, the transformed-data MLE method was also applied in credit risk analysis by Ericsson and Reneby (2004), Wong and Choi (2004) and Duan et al. (2004). The latter derive maximum likelihood estimators for parameters under deterministic and stochastic interest rates. Under Longstaff and Schwartz's (1995) model they propose a two-stage estimation procedure that first analyzes a reduced version of the model by setting the interest rate to a constant. Finally, they address the full version of Longstaff and Schwartz's (1995) structural model. The second estimation methodology KMV, as is called in the financial industry, is based on an iterated algorithm. Interestingly, Duan et al. (2005) proved that the KMV method is somewhat equivalent to the transformed-data MLE method that he proposed in earlier research (Duan, 1994, 2000). As a future research avenue, we will conduct an empirical analysis based on our valuation algorithm and oppose it to real data.

2.5 Conclusion

We propose a general model for valuing risky corporate debt that incorporates both default risk and interest-rate risk. Our methodology is based on a dynamic program coupled with piecewise bilinear approximations where we use parallel computing to enhance efficiency. The proposed model allows for any debt structure with different seniority classes and takes into account tax benefits and bankruptcy costs. Our methodology is flexible and general, and can easily be used to perform realistic empirical credit-risk studies.

We examine the theoretical effect of interest rate uncertainty on the valuation of corporate debt by incorporating a mean-reverting process to model the short-term interest rate. As expected, our results are consistent with empirical evidence documented in the literature. In fact, the interest-rate risk affects the credit spreads' level, and both are negatively correlated. In addition, the correlation between the interest rate and the firms' economic activities explains the observed different credit spreads for bonds with the same rating but in various industries.

Future research avenues include considering a reorganization process for this framework, and the valuation of options embedded in corporate bonds, such as exchangeable convertible bonds. Moreover, one can extend this two-dimensional dynamic program to higher dimensions by including an additional factor to the valuation framework. For example, one could consider a corporate debt for which coupon payments are due in a foreign currency; then, the exchange rate thus becomes the third factor of the model. The extension is challenging but feasible as we can rely on parallel computing to control the computing times, and we can combine the dynamic program with quasi-Monte Carlo simulations instead of finite elements.

APPENDIX

2.A Forward measure

The forward measure $\mathbb{P}^{T_F}$ for any date T_F is the measure associated with taking the bond $B(t,T_F)$ as a numeraire asset. Under the forward measure, the ratio $B(t,T)/B(t,T_F)$ is a martingale for $T \leq T_F$. From Girsanov's Theorem, it follows that the process W^{T_F} defined by

$$dW_t^{T_F} = dZ_t^1 + \frac{\sigma_r}{\alpha}(1-e^{-\alpha(T_F-t)}),$$

is standard Brownian motion under $\mathbb{P}_{T_F}$. Thus, the dynamic of the interest rate becomes

$$dr_t = \left(\theta - \alpha r_t - \frac{\sigma_r^2}{\alpha}(1-e^{-\alpha(T_F-t)})\right)dt + \sigma_r dW_t^{T_F},$$

with $\theta = \alpha\beta$ and the dynamic of $X_t = \ln(V_t)$ is

$$dX_t = \left(r_t - \delta - \frac{\sigma_V^2}{2} - \frac{\rho\sigma_V\sigma_r}{\alpha}(1-e^{-\alpha(T_F-t)})\right)dt + \sigma_V\left(\rho dW_t^{T_F} + \sqrt{1-\rho^2}dZ_t^2\right).$$

The solutions are given by

$$\begin{aligned} r_t &= r_u e^{-\alpha(t-u)} + \left(\frac{\theta}{\alpha} - \frac{\sigma_r^2}{\alpha^2}\right)\left(1-e^{-\alpha(t-u)}\right) + \frac{\sigma_r^2}{2\alpha^2}\left(e^{-\alpha(T_F-t)} - e^{-\alpha(T_F+t-2u)}\right) + \\ &\quad \sigma_r\int_u^t e^{-\alpha(t-s)}dW_s^{T_F}, \\ X_t &= X_u + \beta(u,t) - \left(\frac{\sigma_V^2}{2} + \frac{\rho\sigma_V\sigma_r}{\alpha}\right)(t-u) + \frac{\rho\sigma_V\sigma_r}{\alpha^2}\left(e^{-\alpha(T_F-u)} - e^{-\alpha(T_F-u)}\right) + \\ &\quad \int_u^t\left(\rho\sigma_V + \frac{\sigma_r}{\alpha}\left(1-e^{-\alpha(t-s)}\right)\right)dW_s^{T_F}, \quad \text{for } 0 \leq u \leq t, \end{aligned}$$

with

$$\begin{aligned}\beta(u,t) &= \frac{r_u}{\alpha}\left(1-e^{-\alpha(t-u)}\right)+\left(\frac{\theta}{\alpha}-\frac{\sigma_r^2}{\alpha^2}\right)\left(-\frac{1-e^{-\alpha(t-u)}}{\alpha}+t-u\right)+\\ &\quad \frac{\sigma_r^2}{2\alpha^3}\left(e^{-\alpha(T_F-t)}-2e^{-\alpha(T_F-u)}+e^{-\alpha(T_F+t-2u)}\right).\end{aligned}$$

Under the forward measure, the pair (X_t, r_t) follows a bivariate normal distribution with

$$\begin{aligned}E[X_t|X_u] &= X_u+\beta(u,t)-\left(\frac{\sigma_V^2}{2}+\frac{\rho\sigma_V\sigma_r}{\alpha}\right)(t-u)+\\ &\quad \frac{\rho\sigma_V\sigma_r}{\alpha^2}\left(e^{-\alpha(T_F-t)}-e^{-\alpha(T_F-u)}\right),\\ Var[X_t|X_u] &= \left(\sigma_V^2+\frac{2\rho\sigma_V\sigma_r}{\alpha}+\frac{\sigma_r^2}{\alpha^2}\right)(t-u)-\frac{2\rho\sigma_V\sigma_r}{\alpha^2}\left(1-e^{-\alpha(t-u)}\right)-\\ &\quad \frac{\sigma_r^2}{2\alpha^3}\left(3-4e^{-\alpha(t-u)}+e^{-2\alpha(t-u)}\right),\\ E[r_t|r_u] &= r_u e^{-\alpha(t-u)}+\frac{\theta}{\alpha}\left(1-e^{-\alpha(t-u)}\right)-\frac{\sigma_r^2}{\alpha^2}\left(1-e^{-\alpha(t-u)}\right)+\\ &\quad \frac{\sigma_r^2}{2\alpha^2}\left(e^{-\alpha(T_F-t)}-e^{-\alpha(T_F+t-2u)}\right),\\ Var[r_t|r_u] &= \frac{\sigma_r^2}{2\alpha}\left(1-e^{-2\alpha(t-u)}\right),\\ Cov[X_t,r_t|X_u,r_u] &= \left(\frac{\rho\sigma_V\sigma_r}{\alpha}+\frac{\sigma_r^2}{\alpha^2}\right)\left(1-e^{-\alpha(t-u)}\right)-\frac{\sigma_r^2}{2\alpha^2}\left(1-e^{-2\alpha(t-u)}\right).\end{aligned}$$

2.B Transitions parameters

The transition parameters $T^{\nu\mu}_{klij}$ for ν and $\mu \in \{0,1\}$, $k \in \{1,\ldots,p\}$, $l \in \{1,\ldots,q\}$, $i \in \{0,\ldots,p\}$, and $j \in \{0,\ldots,q\}$ are calculated as follows:

$$
\begin{aligned}
T^{00}_{klij} &= \mathbb{E}^*\left[\mathbb{I}\left((V_{t_{n+1}}, r_{t_{n+1}}) \in R_{ij}\right) \mid (V_{t_n}, r_{t_n}) = (a_k, b_l)\right] \\
&= \mathbb{Q}^*\left[(V_{t_{n+1}}, r_{t_{n+1}}) \in R_{ij} \mid (V_{t_n}, r_{t_n}) = (a_k, b_l)\right] \\
&= \int_{x_{k,i}}^{x_{k,i+1}} \int_{y_{l,j}}^{y_{l,j+1}} \phi(z_1, z_2, \rho) dz_1 dz_2 \\
&= \Phi(x_{k,i+1}, y_{l,j+1}, \rho) - \Phi(x_{k,i}, y_{l,j+1}, \rho) - \Phi(x_{k,i+1}, y_{l,j}, \rho) + \Phi(x_{k,i}, y_{l,j}, \rho),
\end{aligned}
$$

where

$$
\begin{aligned}
x_{k,i} &= \left(\log\left(a_i/a_k\right) - \eta_1\right)/\sqrt{\delta_1} \\
y_{l,j} &= \left(b_j - \eta_2\right)/\sqrt{\delta_2}, \\
\eta_1 &= \beta_l - \left(\frac{\sigma_V^2}{2} + \frac{\rho\sigma_V\sigma_r}{\alpha}\right)\Delta t + \frac{\rho\sigma_V\sigma_r}{\alpha^2}\left(1 - e^{-\alpha\Delta t}\right), \\
\delta_1 &= \left(\sigma_V^2 + \frac{2\rho\sigma_V\sigma_r}{\alpha} + \frac{\sigma_r^2}{\alpha^2}\right)\Delta t - \frac{2\rho\sigma_V\sigma_r}{\alpha^2}\left(1 - e^{-\alpha\Delta t}\right) - \\
&\quad \frac{\sigma_r^2}{2\alpha^3}\left(3 - 4e^{-\alpha\Delta t} + e^{-2\alpha\Delta t}\right), \\
\eta_2 &= b_l e^{-\alpha\Delta t} + \frac{\theta}{\alpha}\left(1 - e^{-\alpha\Delta t}\right) - \frac{\sigma_r^2}{\alpha^2}\left(1 - e^{-\alpha\Delta t}\right) + \frac{\sigma_r^2}{2\alpha^2}\left(1 - e^{-2\alpha\Delta t}\right), \\
\delta_2 &= \frac{\sigma_r^2}{2\alpha}\left(1 - e^{-2\alpha\Delta t}\right), \\
\beta_l &= \frac{r_l}{\alpha}\left(1 - e^{-\alpha\Delta t}\right) + \left(\frac{\theta}{\alpha} - \frac{\sigma_r^2}{\alpha^2}\right)\left(-\frac{1 - e^{-\alpha\Delta t}}{\alpha} + \Delta t\right) \\
&\quad + \frac{\sigma_r^2}{2\alpha^3}\left(1 - 2e^{-\alpha\Delta t} + e^{-2\alpha\Delta t}\right).
\end{aligned}
$$

$\mathbb{E}^*$ is the expectation under the forward measure to the time t_{n+1}. The functions $\phi(\cdot,\cdot,\rho)$ and $\Phi(\cdot,\cdot,\rho)$ are the density and cumulative density functions, respectively, of the bivariate standard normal distribution with correlation coefficient ρ. The function $\Phi(\cdot,\cdot,\rho)$ is computed according to Genz (2004).

$$
\begin{aligned}
T^{10}_{klij} &= \mathbb{E}^*\left[V_{t_{n+1}}\mathbb{I}\left((V_{t_{n+1}},r_{t_{n+1}})\in R_{ij}\right)\mid (V_{t_n},r_{t_n})=(a_k,b_l)\right]\\
&= w^1_k\int_{x_{k,i}-\sqrt{\delta_1}}^{x_{k,i+1}-\sqrt{\delta_1}}\int_{y_{l,j}-\rho\sqrt{\delta_1}}^{y_{l,j+1}-\rho\sqrt{\delta_1}}\phi(u_1,u_2,\rho)du_1du_2\\
&= w^1_k\Big[\Phi(x_{k,i+1}-\sqrt{\delta_1},y_{l,j+1}-\rho\sqrt{\delta_1},\rho)-\Phi(x_{k,i}-\sqrt{\delta_1},y_{l,j+1}-\rho\sqrt{\delta_1},\rho)\\
&\quad -\Phi(x_{k,i+1}-\sqrt{\delta_1},y_{l,j}-\rho\sqrt{\delta_1},\rho)+\Phi(x_{k,i}-\sqrt{\delta_1},y_{l,j}-\rho\sqrt{\delta_1},\rho)\Big],
\end{aligned}
$$

where $w^1_k = a_k\exp\left(\eta_1+\delta_1/2\right)$.

$$
\begin{aligned}
T^{01}_{klij} &= \mathbb{E}^*\left[e^{r_{t_{n+1}}}\mathbb{I}\left((V_{t_{n+1}},r_{t_{n+1}})\in R_{ij}\right)\mid (V_{t_n},r_{t_n})=(a_k,b_l)\right]\\
&= w^2_l\int_{x_{k,i}-\rho\sigma_2\Delta t}^{x_{k,i+1}-\rho\sigma_2\Delta t}\int_{y_{l,j}-\sigma_2\Delta t}^{y_{l,j+1}-\sigma_2\Delta t}\phi(u_1,u_2,\rho)du_1du_2\\
&= w^2_l\Big[\Phi(x_{k,i+1}-\rho\sqrt{\delta_2},y_{l,j+1}-\sqrt{\delta_2},\rho)-\Phi(x_{k,i}-\rho\sqrt{\delta_2},y_{l,j+1}-\sqrt{\delta_2},\rho)\\
&\quad -\Phi(x_{k,i+1}-\rho\sqrt{\delta_2},y_{l,j}-\sqrt{\delta_2},\rho)+\Phi(x_{k,i}-\rho\sqrt{\delta_2},y_{l,j}-\sqrt{\delta_2},\rho)\Big],
\end{aligned}
$$

where $w^2_l = \exp\left(\eta_2+\delta_2/2\right)$.

$$
\begin{aligned}
T^{11}_{klij} &= \mathbb{E}^* \left[V_{t_{n+1}} e^{r_{t_{n+1}}} \mathbb{I}\left((V_{t_{n+1}}, r_{t_{n+1}}) \in R_{ij} \right) \mid (V_{t_n}, r_{t_n}) = (a_k, b_l) \right] \\
&= w^1_k w^2_l \exp\left(\rho\sqrt{\delta_1\delta_2}\right) \int_{x_{k,i}-\sqrt{\delta_1}-\rho\sqrt{\delta_2}}^{x_{k,i+1}-\sqrt{\delta_1}-\rho\sqrt{\delta_2}} \int_{y_{l,j}-\rho\sqrt{\delta_1}-\sqrt{\delta_2}}^{y_{l,j+1}-\rho\sqrt{\delta_1}-\sqrt{\delta_2}} \phi(u_1,u_2,\rho) du_1 du_2 \\
&= w^1_k w^2_l \exp\left(\rho\sqrt{\delta_1\delta_2}\right) \Big[\Phi(x_{k,i+1} - \sqrt{\delta_1} - \rho\sqrt{\delta_2}, y_{l,j+1} - \rho\sqrt{\delta_1} - \sqrt{\delta_2}, \rho) - \\
&\quad \Phi(x_{k,i} - \sqrt{\delta_1} - \rho\sqrt{\delta_2}, y_{l,j+1} - \rho\sqrt{\delta_1} - \sqrt{\delta_2}, \rho) - \\
&\quad \Phi(x_{k,i+1} - \sqrt{\delta_1} - \rho\sqrt{\delta_2}, y_{l,j} - \rho\sqrt{\delta_1} - \sqrt{\delta_2}, \rho) + \\
&\quad \Phi(x_{k,i} - \sqrt{\delta_1} - \rho\sqrt{\delta_2}, y_{l,j} - -\rho\sqrt{\delta_1} - \sqrt{\delta_2}, \rho) \Big].
\end{aligned}
$$

2.C Parallel computing algorithm

Parallel computing uses multiple central processing units (CPUs) simultaneously to speed-up complex computations. The Message Passing Interface (MPI) library allows the computing process to exchange information between the running CPU environments in order to achieve a given job. Each CPU has access to a certain memory space. MPI requires case-sensitive programming changes from the serial code to its parallel version.

The easiest way to parallelize DP is to submit the computation tasks associated to a given grid point (a_k, b_l), for $k = 1, \ldots, p$ and $l = 1, \ldots, q$, to a single CPU. Our parallel code acts as follows.

1. This single CPU computes once and locally stores the overall grid points (a_i, b_j) and each value function values $\Psi_N(a_i, b_j)$, for $i = 1, \ldots, p$ and $j = 1, \ldots, q$.
2. It also computes once and locally stores the $4 \times (p+1)(q+1)$ transition parameters T^{00}_{klij}, T^{10}_{klij}, T^{01}_{klij}, and T^{11}_{klij}, for $i = 0, \ldots, p$ and $j = 0, \ldots, q$.

3. It computes and stores at step $n+1$ the local coefficients α_{ij}^{n+1}, β_{ij}^{n+1}, γ_{ij}^{n+1}, and δ_{ij}^{n+1}, for each value function Ψ_{n+1}, for $i=0,\ldots,p$ and $j=0,\ldots,q$.
4. It computes and stores at step n every value function $\tilde{\Psi}_n(a_k,b_l)$.
5. The same CPU exports $\tilde{\Psi}_n(a_k,b_l)$ to a selected CPU, the so-called master CPU.
6. The master CPU collects $\tilde{\Psi}_n(a_k,b_l)$, for $k=1,\ldots,p$ and $l=1,\ldots,q$, and sends them back to all running CPUs.
7. Go to step 3 and repeat until $n=0$.

Since the number of CPUs available to the analyst is usually less than the grid size pq, we submit the same number of grid points to each CPU.

BIBLIOGRAPHY

Altieri, A. and Vargiolu, T. (2001). Optimal default boundary in discrete time models. *Rendiconti per gli Studi Economici Quantitativi*, 2(1):1–20.

Anderson, R. and Sundaresan, S. (1996). Design and valuation of debt contracts. *Review of Financial Studies*, 9(1):37–68.

Ayadi, M. A., Ben-Ameur, H., and Fakhfakh, T. (2016). A dynamic program for valuing corporate securities. *European Journal of Operational Research*, 249(2):751–770.

Ben-Abdellatif, M., Rémillard, B., and Chérif, R. (2016). Valuing corporate securities. Working paper, HEC Montréal.

Black, F. and Cox, J. C. (1976). Valuing corporate securities: some effects of bond indenture provisions. *The Journal of Finance*, 31(2):351–367.

Briys, E. and De Varenne, F. (1997). Valuing risky fixed rate debt: an extension. *Journal of Financial and Quantitative Analysis*, 32(02):239–248.

Cathcart, L. and El-Jahel, L. (1998). Valuation of defaultable bonds. *The Journal of Fixed Income*, 8(1):65–78.

Chen, N. and Kou, S. G. (2009). Credit spreads, optimal capital structure, and implied volatility with endogenous default and jump risk. *Mathematical Finance*, 19(3):343–378.

Collin-Dufresne, P. and Goldstein, R. S. (2001). Do credit spreads reflect stationary leverage ratios? *The Journal of Finance*, 56(5):1929–1957.

Collin-Dufresne, P., Goldstein, R. S., and Martin, J. S. (2001). The determinants of credit spread changes. *The Journal of Finance*, 56(6):2177–2207.

Cox, J. C., Ingersoll Jr, J. E., and Ross, S. A. (1985). A theory of the term structure of interest rates. *Econometrica: Journal of the Econometric Society*, 53:385–407.

Duan, J.-C. (1994). Maximum likelihood estimation using price data of the derivative contract. *Mathematical Finance*, 4(2):155–167.

Duan, J.-C. (2000). Correction: Maximum likelihood estimation using price data of the derivative contract (mathematical finance 1994, 4/2, 155–167). *Mathematical Finance*, 10(4):461–462.

Duan, J.-C., Gauthier, G., and Simonato, J.-G. (2005). *On the equivalence of the KMV and maximum likelihood methods for structural credit risk models*. Groupe d'études et de recherche en analyse des décisions.

Duan, J.-C., Simonato, J.-G., Gauthier, G., and Zaanoun, S. (2004). Estimating merton's model by maximum likelihood with survivorship consideration. In *EFA 2004 Maastricht Meetings Paper*, number 4190.

Ericsson, J. and Reneby, J. (1998). A framework for valuing corporate securities. *Applied Mathematical Finance*, 5(3-4):143–163.

Ericsson, J. and Reneby, J. (2004). An empirical study of structural credit risk models using stock and bond prices. *The Journal of Fixed Income*, 13(4):38–49.

François, P. and Morellec, E. (2004). Capital structure and asset prices: some effects of bankruptcy procedures. *The Journal of Business*, 77(2):387–411.

Genz, A. (2004). Numerical computation of rectangular bivariate and trivariate normal and t probabilities. *Statistics and Computing*, 14(3):251–260.

Geske, R. (1977). The valuation of corporate liabilities as compound options. *The Journal of Financial and Quantitative Analysis*, 12(4):541–552.

Hsu, J., Saá-Requejo, J., and Santa-Clara, P. (2010). A structural model of default risk. *Journal of Fixed Income*, 19(3):77–94.

Jamshidian, F. (1989). An exact bond option formula. *The journal of Finance*, 44(1):205–209.

Kim, I. J., Ramaswamy, K., and Sundaresan, S. (1993). Does default risk in coupons affect the valuation of corporate bonds?: A contingent claims model. *Financial Management*, 22(1):117–131.

Leland, H. E. (1994). Corporate debt value, bond covenants, and optimal capital structure. *The Journal of Finance*, 49(4):1213–1252.

Leland, H. E. and Toft, K. B. (1996). Optimal capital structure, endogenous bankruptcy, and the term structure of credit spreads. *The Journal of Finance*, 51(3):987–1019.

Litterman, R. B. and Scheinkman, J. (1991). Common factors affecting bond returns. *The Journal of Fixed Income*, 1(1):54–61.

Longstaff, F. A. and Schwartz, E. S. (1995). A simple approach to valuing risky fixed and floating rate debt. *The Journal of Finance*, 50(3):789–819.

Mella-Barral, P. and Perraudin, W. (1997). Strategic debt service. *The Journal of Finance*, 52(2):531–556.

Merton, R. C. (1974). On the pricing of corporate debt: the risk structure of interest rates. *The Journal of Finance*, 29(2):449–470.

Sarig, O. and Warga, A. (1989). Some empirical estimates of the risk structure of interest rates. *The Journal of Finance*, 44(5):1351–1360.

Shimko, D. C., Tejima, N., and Van Deventer, D. R. (1993). The pricing of risky debt when interest rates are stochastic. *The Journal of Fixed Income*, 3(2):58–65.

Vasicek, O. (1977). An equilibrium characterization of the term structure. *Journal of Financial Economics*, 5(2):177–188.

Wong, H. and Choi, T. (2004). The impact of default barrier on the market value of firm's asset. Working paper, Chinese University of Hong Kong.

Zhou, C. (2001). The term structure of credit spreads with jump risk. *Journal of Banking and Finance*, 25(11):2015–2040.

CHAPTER 3

A STRUCTURAL MODEL FOR VALUING EXCHANGEABLE BONDS

Malek Ben-Abdellatif, [1] Hatem Ben Ameur, [2] Bruno Rémillard [3]

Abstract

An exchangeable bond is a debt that is convertible into shares of a firm's equity other than the bond's issuer. We evaluate an exchangeable bond within a two-dimensional structural model, where the assets' value of the bond's issuer and the underlying equity value are the state variables. Our model, based on dynamic programming, finite elements, and parallel computing, accommodates arbitrary debt portfolio including an exchangeable bond, several seniority classes, bankruptcy costs and tax benefits. We conduct a numerical investigation that highlights the main characteristics of exchangeable bonds and their distinction from a straight bond.

Keywords: Credit risk; Structural model; Exchangeable bond; Dynamic programming; Finite elements; Parallel computing.

3.1 Introduction

The main aim of this paper is to value exchangeable bonds. Unlike a convertible debt whose the payoff is associated with the performance of the issuer's stock, the payoff of an exchangeable debt depends on the stock of a different firm. Specifically,

1. Malek Ben-Abdellatif is a Ph.D. student at HEC Montréal.
2. Hatem Ben-Ameur is a Professor at HEC Montréal in the Department of Decision Sciences, and member of the GERAD.
3. Bruno Rémillard is a Professor at HEC Montréal in the Department of Decision Sciences, and member of the CRM and GERAD.

Partial funding in support of this work was provided by the Natural Sciences and Engineering Research Council of Canada, the Fonds pour la formation de chercheurs et l'aide à la recherche du Gouvernement du Québec, and the Institut de Finance Mathématique du Québec.

a firm that issues an exchangeable debt gives bondholders the option to exchange their bonds for shares of another firm's equity. The exercise decision is also closely related to the issuer's financial situation, in particular its credit risk default.

Exchangeable debt has been offered by firms since the early 1970s. The Association of Convertible Bonds Management reported that, in 2001, about one third of the European convertible bond market was made of exchangeable bonds. According to Grimwood and Hodges (2002), this proportion represents 14% of the total bond market in the US.

Assume a company holds equity shares of another public company and makes the decision to divest of this intercorporate holding because of negative expectations regarding its future prospects. Divesting strategies include block sales, secondary distributions or issuing exchangeable debt. As documented in corporate finance (Barber, 1993), the latter is preferred over the other two alternatives. In fact, announcing a secondary distribution (Mikkelson and Partch, 1985) or block sales (Holthausen et al., 1987) can provoke a negative price reaction, which can be avoided by issuing an exchangeable debt. Jones and Mason (1986) discuss also tax advantages as motivation for exchangeable debt issues.

Many articles address the valuation of ordinary convertible bonds, see, e.g. Ingersoll (1977) and Brennan and Schwartz (1977), to cite a few. Despite the relevance of exchangeable debts, less attention has been given to their theoretical valuation. Realdon (2004) proposes a structural valuation model for these bonds and uses the Hopscotch finite difference method to solve the problem. He considers the case of an exchangeable bond when the issuer owns the underlying shares and when the issuer does not own these shares. He explains however that the latter case is not realistic. He also discusses some features and distinctions of the exchangeable bond. Moreover, Guo and Ren (2009) present a pricing model for exchangeable debt under the least-squares regression approach proposed by Longstaff and Schwartz (2001).

In this paper, we extend Realdon (2004) by presenting a two-factor structural model and incorporating the exchangeable debt as part of the debt portfolio of the firm under a setting comparable to Ayadi et al. (2016). Our model accounts for tax benefits, bankruptcy costs and an arbitrary debt portfolio, allowing our model to be flexible and able to accommodate any financial structural. The two factors are the value of the issuer's assets and the value of the equity shares against which the bond can be exchanged. Our methodology is based on a two-dimensional dynamic program coupled with bilinear interpolations and parallel computing. We suppose that the issuer owns the shares of the underlying equity, which are pledged to the bondholders of the exchangeable bond. This is to ensure that the exchange option is not lost in case of default. We start the evaluation at maturity of the debt where we can assess the debt in closed form. Next we proceed backward and evaluate the bond at every payment date. On the one hand, the firm survives in each step if it can meet its financial commitments to pay coupons and principal amounts to the bondholders. In this case, bondholders of exchangeable bond will compare what they receive to the value of the underlying shares, and exercise the exchange option if it is beneficial. If the option is exercised, the firm again reassess its situation: the total value drops if it no longer owns the underlying shares, and default occurs if senior bondholders cannot be paid. On the other hand, the firm defaults if it cannot honor its commitments to the bondholders. Those of the exchangeable bond will then compare their recovered amount to the value of the underlying shares and exercise, if favorable to them. Upon exercise, the firm can still survive if it can pay the senior bondholders and avoid default.

The paper is organized as follows: Section 3.2 presents our valuation framework, Section 3.3 describes our dynamic program, Section 3.4 shows our numerical investigation, and Section 3.5 concludes our paper.

3.2 Valuation framework

The issuer credit risk is modeled using a structural model. We consider that the assets' value V_t moves according to geometric-Brownian motion

$$\frac{dV_t}{V_t} = (r - \delta_1)dt + \sigma_V dW_t^1,$$

where r is the constant risk-free rate, δ_1 is the payout rate and σ_V is its volatility. The capital structure of the issuer contains a portfolio of a straight debt and an exchangeable debt, as well as a common stock. The straight debt is a senior debt. The firm is committed to making coupon payments to the bondholders which results in collecting tax benefits. Let $\mathscr{P} = \{t_0, t_1, \ldots, t_n, \ldots, t_N\}$ be a set of payment dates. At each date t_n, the firm is committed to pay $d_n = d_n^e + d_n^{\bar{e}}$ to its creditors, where d_n^e and $d_n^{\bar{e}}$ are the payments due to the bondholders of the exchangeable bond and to the other bondholders respectively. These payments include principal as well as coupon payments. The interest payments are noted C_n^e and $C_n^{\bar{e}}$ respectively. The last payment dates for the debts are indicated by $T^{\bar{e}} \leq T^e = T$. The tax benefits at each payment date t_n are denoted by $tb_n = tb_n^e + tb_n^{\bar{e}}$ where $tb_n^e = r^c C_n^e$, $tb_n^{\bar{e}} = r^c C_n^{\bar{e}}$, and $r^c \in [0, 1]$ is the corporate tax rate. The firm also pays bankruptcy costs in case of default proportional to the remaining assets' value, i.e. wV, where $w \in [0, 1]$ is a constant fraction.

The model assumes that the stockholders determine the time of default by maximizing the firm's total value subject to the limited liability constraint. In addition, the bondholders of the exchangeable debt have the possibility to exchange their bond for a set number of another company's shares at any date until maturity and in case of default. These shares move according to geometric-Brownian motion as follows:

$$\frac{dS_t}{S_t} = (r - \delta_2)dt + \sigma_S dW_t^2,$$

where δ_2 is a continuous dividend rate, σ_S is the shares' volatility, and $(W^1,\ W^2)$

is a bivariate correlated Brownian motion with

$$\mathrm{Cor}(W_t^1, W_t^2) = \rho, \quad \text{for all } t > 0.$$

We suppose that the shares are pledged to the bondholders of the exchangeable bond, which prevents the exchange option from being lost. We assume the strict priority rule under default. The non-exchangeable bondholders are paid before the exchangeable bondholders unless the later exercise their right. We also suppose that the shares underlying the exchangeable bond are protected against bankruptcy costs.

The balance-sheet equality at time t_n is then

$$v + s + TB_n(v,s) - BC_n(v,s) = D_n^{\bar{e}}(v,s) + D_n^e(v,s) + \mathscr{E}_n(v,s), \tag{3.1}$$

where $V_n = v$ and $S_n = s$. The functions $TB_n(v,s)$ and $BC_n(v,s)$ are the value of the tax benefits and the value of the bankruptcy costs at date t_n, respectively. $D_n^{\bar{e}}(v,s)$ is the value of the straight bond, $D_n^e(v,s)$ is the value of the exchangeable bond, and $\mathscr{E}_n(v,s)$ is the equity value of the issuer at date t_n. These corporate securities are seen as financial derivatives on the firm's assets value and the exchangeable bond's underlying shares.

At each payment/decision date, several scenarios can happen, depending on the exchange option holder's decision (holding/exercise) and the firm's status (survival/default). We indicate by $F^{e+\bar{e}}$ the firm under holding with its overall exchangeable and non-exchangeable debt and by $F^{\bar{e}}$ the firm just after exercise with its remaining non-exchangeable debt. The scenarios at maturity are as follows:

Case 1: Holding under survival

The holding condition is

$$s \leq D_N^e(v,s) = d_N^e.$$

The balance-sheet equality of $F^{e+\bar{e}}$ is

$$v+s+tb_N-0=d_N^e+d_N^{\bar{e}}+(v+s+tb_N-d_N),$$

which results in the survival condition

$$v+s+tb_N-d_N>0.$$

The value functions are

$$\begin{aligned}
TB_N(v,s)&=tb_N,\\
BC_N(v,s)&=0,\\
D_N^{\bar{e}}(v,s)&=d_N^{\bar{e}},\\
D_N^{e}(v,s)&=d_N^{e},\\
\mathscr{E}_N(v,s)&=v+s+tb_N-d_N.
\end{aligned}$$

Case 2: Holding under default

The default condition of $F^{e+\bar{e}}$ is

$$v+s+tb_N-d_N\leq 0,$$

as explained in case 1, while its balance-sheet equality is

$$v+s+0-wv=\min((1-w)v+s,d_N^{\bar{e}})+\max((1-w)v+s-d_N^{\bar{e}},0)+0,$$

and the value functions are

$$
\begin{aligned}
TB_N(v,s) &= 0,\\
BC_N(v,s) &= wv,\\
D_N^{\bar{e}}(v,s) &= \min((1-w)v+s, d_N^{\bar{e}}),\\
D_N^{e}(v,s) &= \max((1-w)v+s-d_N^{\bar{e}}, 0),\\
\mathscr{E}_N(v,s) &= 0,
\end{aligned}
$$

which result in the holding condition

$$s \leq D_N^{e}(v,s) = \max((1-w)v+s-d_N^{\bar{e}}, 0)$$

All in all, one has

$$
\begin{aligned}
D_N^{\bar{e}}(v,s) =& d_N^{\bar{e}},\\
D_N^{e}(v,s) =& (1-w)v+s-d_N^{\bar{e}},
\end{aligned}
$$

The straight bondholders are fully paid and the exchangeable bondholders are partially paid, while exercising the option is suboptimal.

Case 3: Exercise: $F^{e+\bar{e}}$ and $F^{\bar{e}}$ survive

The survival condition of $F^{e+\bar{e}}$ is

$$v+s+tb_N-d_N > 0,$$

as explained in case 1, while the exercise condition is

$$D_N^{e}(v,s) = s > d_N^{e}.$$

After exercise, the exchangeable debt no longer belongs to the firm's debt portfolio.

The balance-sheet equality for the $F^{\bar{e}}$ becomes

$$v + tb_N^{\bar{e}} - 0 = d_N^{\bar{e}} + (v + tb_N^{\bar{e}} - d_N^{\bar{e}})$$

The survival condition of $F^{\bar{e}}$ is then

$$v + tb_N^{\bar{e}} - d_N^{\bar{e}} > 0.$$

The value functions are

$$\begin{aligned}
TB_N(v,s) &= tb_N^{\bar{e}}, \\
BC_N(v,s) &= 0, \\
D_N^{\bar{e}}(v,s) &= d_N^{\bar{e}}, \\
D_N^{e}(v,s) &= s, \\
\mathscr{E}_N(v,s) &= v + tb_N^{\bar{e}} - d_N^{\bar{e}}.
\end{aligned}$$

Case 4: Exercise: $F^{e+\bar{e}}$ survives and $F^{\bar{e}}$ defaults

The survival condition of $F^{e+\bar{e}}$ is

$$v + s + tb_N - d_N > 0,$$

as explained in case 1, while the exercise condition is

$$D_N^{e}(v,s) = s > d_N^{e}.$$

After exercise, the exchangeable debt no longer belongs to the firm's debt portfolio. The firm $F^{\bar{e}}$ defaults if

$$v + tb_N^{\bar{e}} - d_N^{\bar{e}} \leq 0,$$

as explained in case 3, and its balance-sheet equality becomes

$$v + 0 - wv = (1 - w)v + 0.$$

It's worth noticing that $d_N^{\bar{e}} \geq v + tb_N^{\bar{e}} \geq v \geq (1-w)v$. The value functions are

$$\begin{aligned} TB_N(v,s) &= 0, \\ BC_N(v,s) &= wv, \\ D_N^{\bar{e}}(v,s) &= (1-w)v, \\ D_N^{e}(v,s) &= s, \\ \mathscr{E}_N(v,s) &= 0. \end{aligned}$$

Exchanging the bond provokes default.

Case 5: Exercise: $F^{e+\bar{e}}$ defaults and $F^{\bar{e}}$ survives

The firm $F^{e+\bar{e}}$ would have defaulted if the exchange option has not been exercised. The default condition is

$$v + s + tb_N - d_N \leq 0.$$

and the exercise condition is

$$D_N^{e}(v,s) = s > (1-w)v + s - d_N^{\bar{e}},$$

as explained in case 2. After exercise, the exchangeable debt no longer belongs to the firm's debt portfolio. The balance-sheet equality of $F^{\bar{e}}$ becomes

$$v + tb_N^{\bar{e}} + 0 = d_N^{\bar{e}} + (v + tb_N^{\bar{e}} - d_N^{\bar{e}}),$$

and the survival condition is

$$v + tb_N^{\bar{e}} - d_N^{\bar{e}} > 0.$$

The value functions are

$$\begin{aligned} TB_N(v,s) &= tb_N^{\bar{e}}, \\ BC_N(v,s) &= 0, \\ D_N^{\bar{e}}(v,s) &= d_N^{\bar{e}}, \\ D_N^{e}(v,s) &= s, \\ \mathscr{E}_N(v,s) &= v + tb_N^{\bar{e}} - d_N^{\bar{e}}. \end{aligned}$$

Exercising the exchange option prevents the firm from default.

Case 6: Exercise: $F^{e+\bar{e}}$ and $F^{\bar{e}}$ default

The default condition of $F^{e+\bar{e}}$ is

$$v + s + tb_N - d_N \leq 0.$$

and the exercise condition is

$$D_N^e = s > (1-w)v + s - d_N^{\bar{e}},$$

as explained in case 2. After exercise, the exchangeable debt no longer belongs to the firm's debt portfolio. The firm $F^{\bar{e}}$ defaults if

$$v + tb_N^{\bar{e}} - d_N^{\bar{e}} \leq 0,$$

as explained in case 5, and its balance-sheet equality becomes

$$v + 0 - wv = (1-w)v + 0.$$

The value functions are

$$
\begin{aligned}
TB_N(v,s) &= 0, \\
BC_N(v,s) &= wv, \\
D_N^{\bar{e}}(v,s) &= (1-w)v, \\
D_N^{e}(v,s) &= s, \\
\mathscr{E}_N(v,s) &= 0.
\end{aligned}
$$

At any decision date t_n, we apply a similar reasoning. The firm defaults if it cannot meet its financial commitments. The exchangeable bondholders will exercise the exchange option whenever the value of the underlying shares is greater than the promised payments in case of survival, or the recovered amount in case of default. The scenarios at any date t_n are as follows:

Case 1: Holding under survival

The holding condition is

$$
s \leq D_n^e(v,s) = d_n^e + \mathbb{E}\left[D_{n+1}^e(V_{n+1},S_{n+1})|F_n\right].
$$

The survival condition for $F^{e+\bar{e}}$ is

$$
\mathbb{E}\left[\mathscr{E}_{n+1}(V_{n+1},S_{n+1})|F_n\right] + tb_n - d_n > 0.
$$

The value functions are

$$
\begin{aligned}
TB_n(v,s) &= tb_n + \mathbb{E}\left[TB_{n+1}(V_{n+1},S_{n+1})|F_n\right], \\
BC_n(v,s) &= \mathbb{E}\left[BC_{n+1}(V_{n+1},S_{n+1})|F_n\right], \\
D_n^{\bar{e}}(v,s) &= d_N^{\bar{e}} + \mathbb{E}\left[D_{n+1}^{\bar{e}}(V_{n+1},S_{n+1})|F_n\right], \\
D_n^{e}(v,s) &= d_n^e + \mathbb{E}\left[D_{n+1}^{e}(V_{n+1},S_{n+1})|F_n\right], \\
\mathscr{E}_n(v,s) &= \mathbb{E}\left[\mathscr{E}_{n+1}(V_{n+1},S_{n+1})|F_n\right] + tb_n - d_n.
\end{aligned}
$$

Case 2: Holding under default

The default condition of $F^{e+\bar{e}}$ is

$$\mathbb{E}\left[\mathscr{E}_{n+1}(V_{n+1},S_{n+1})|F_n\right]+tb_n-d_n\leq 0$$

The holding condition is

$$s\leq D_n^e(v,s)=(1-w)v+s-d_n^{\bar{e}}-\mathbb{E}\left[D_{n+1}^{\bar{e}}(V_{n+1},S_{n+1})|F_n\right].$$

The value functions are

$$\begin{aligned}
TB_n(v,s)&=0,\\
BC_n(v,s)&=wv,\\
D_n^{\bar{e}}(v,s)&=d_n^{\bar{e}}+\mathbb{E}\left[D_{n+1}^{\bar{e}}(V_{n+1},S_{n+1})|F_n\right],\\
D_n^{e}(v,s)&=(1-w)v+s-d_n^{\bar{e}}-\mathbb{E}\left[D_{n+1}^{\bar{e}}(V_{n+1},S_{n+1})|F_n\right],\\
\mathscr{E}_n(v,s)&=0.
\end{aligned}$$

Case 3: Exercise: $F^{e+\bar{e}}$ and $F^{\bar{e}}$ survive

The survival condition of $F^{e+\bar{e}}$ is

$$\mathbb{E}\left[\mathscr{E}_{n+1}(V_{n+1},S_{n+1})|F_n\right]+tb_n-d_n>0.$$

while the exercise condition is

$$D_n^e(v,s)=s>d_n^e+\mathbb{E}\left[D_{n+1}^{e}(V_{n+1},S_{n+1})|F_n\right].$$

After exercise, the exchangeable debt no longer belongs to the firm's debt portfolio. The survival condition of $F^{\bar{e}}$ is

$$\mathbb{E}\left[\mathscr{E}_{n+1}(V_{n+1},S_{n+1})|F_n\right]+tb_n^{\bar{e}}-d_n^{\bar{e}}>0.$$

The value functions are

$$
\begin{aligned}
TB_n(v,s) &= tb_n^{\bar{e}} + \mathbb{E}\left[TB_{n+1}(V_{n+1},S_{n+1})|F_n\right], \\
BC_n(v,s) &= \mathbb{E}\left[BC_{n+1}(V_{n+1},S_{n+1})|F_n\right], \\
D_n^{\bar{e}}(v,s) &= d_n^{\bar{e}} + \mathbb{E}\left[D_{n+1}^{\bar{e}}(V_{n+1},S_{n+1})|F_n\right], \\
D_n^{e}(v,s) &= s, \\
\mathscr{E}_n(v,s) &= \mathbb{E}\left[\mathscr{E}_{n+1}(V_{n+1},S_{n+1})|F_n\right] + tb_n^{\bar{e}} - d_n^{\bar{e}}.
\end{aligned}
$$

Case 4: Exercise: $F^{e+\bar{e}}$ survives and $F^{\bar{e}}$ defaults

The survival condition of $F^{e+\bar{e}}$ is

$$
\mathbb{E}\left[\mathscr{E}_{n+1}(V_{n+1},S_{n+1})|F_n\right] + tb_n - d_n > 0.
$$

while the exercise condition is

$$
D_n^{e}(v,s) = s > d_n^{e} + \mathbb{E}\left[D_{n+1}^{e}(V_{n+1},S_{n+1})|F_n\right].
$$

After exercise, the exchangeable debt no longer belongs to the firm's debt portfolio. The default condition of $F^{\bar{e}}$ is

$$
\mathbb{E}\left[\mathscr{E}_{n+1}(V_{n+1},S_{n+1})|F_n\right] + tb_n^{\bar{e}} - d_n^{\bar{e}} \leq 0.
$$

The value functions are

$$
\begin{aligned}
TB_n(v,s) &= 0, \\
BC_n(v,s) &= wv, \\
D_n^{\bar{e}}(v,s) &= (1-w)v, \\
D_n^{e}&(v,s)s, \\
\mathscr{E}_n(v,s) &= 0.
\end{aligned}
$$

Exchanging the bond provokes default.

Case 5: Exercise: $F^{e+\bar{e}}$ defaults and $F^{\bar{e}}$ survives

The firm $F^{e+\bar{e}}$ would have defaulted if the exchange option has not been exercised. The default condition of $F^{e+\bar{e}}$ is

$$\mathbb{E}\left[\mathscr{E}_{n+1}(V_{n+1},S_{n+1})|F_n\right]+tb_n-d_n\leq 0$$

The exercise condition is

$$s=D_n^e(v,s)>(1-w)v+s-d_n^{\bar{e}}-\mathbb{E}\left[D_{n+1}^{\bar{e}}(V_{n+1},S_{n+1})|F_n\right].$$

After exercise, the exchangeable debt no longer belongs to the firm's debt portfolio. The survival condition of $F^{\bar{e}}$ is

$$\mathbb{E}\left[\mathscr{E}_{n+1}(V_{n+1},S_{n+1})|F_n\right]+tb_n^{\bar{e}}-d_n^{\bar{e}}>0.$$

The value functions are

$$\begin{aligned}
TB_n(v,s)&=tb_n^{\bar{e}}+\mathbb{E}\left[TB_{n+1}(V_{n+1},S_{n+1})|F_n\right],\\
BC_n(v,s)&=\mathbb{E}\left[BC_{n+1}(V_{n+1},S_{n+1})|F_n\right],\\
D_n^{\bar{e}}(v,s)&=d_n^{\bar{e}}+\mathbb{E}\left[D_{n+1}^{\bar{e}}(V_{n+1},S_{n+1})|F_n\right],\\
D_n^e(v,s)&=s,\\
\mathscr{E}_n(v,s)&=\mathbb{E}\left[\mathscr{E}_{n+1}(V_{n+1},S_{n+1})|F_n\right]+tb_n^{\bar{e}}-d_n^{\bar{e}}.
\end{aligned}$$

Exercising the exchange option prevents the firm from default.

Case 6: Exercise: $F^{e+\bar{e}}$ and $F^{\bar{e}}$ default

The default condition of $F^{e+\bar{e}}$ is

$$\mathbb{E}\left[\mathscr{E}_{n+1}(V_{n+1},S_{n+1})|F_n\right]+tb_n-d_n\leq 0$$

The exercise condition is

$$s = D^e_n(v,s) > (1-w)v + s - d^{\bar{e}}_n - \mathbb{E}\left[D^{\bar{e}}_{n+1}(V_{n+1},S_{n+1})|F_n\right].$$

After exercise, the exchangeable debt no longer belongs to the firm's debt portfolio. The default condition of $F^{\bar{e}}$ is

$$\mathbb{E}\left[\mathscr{E}_{n+1}(V_{n+1},S_{n+1})|F_n\right] + tb^{\bar{e}}_n - d^{\bar{e}}_n \leq 0.$$

The value functions are

$$\begin{aligned} TB_n(v,s) &= 0, \\ BC_n(v,s) &= wv, \\ D^{\bar{e}}_n(v,s) &= (1-w)v, \\ D^e_n(v,s) &= s, \\ \mathscr{E}_n(v,s) &= 0. \end{aligned}$$

This model can be studied under the assumption that the issuer of the exchangeable bond does not own the equity shares against which the debt can be exchanged. In this case, if the bondholders decide to exercise the option, the firm has to purchase the shares to deliver them. The exchange option can thus be lost if the firm is in distress and cannot deliver the shares. Under this hypothesis, the issuer's default probability increases and the exchangeable bond is less valuable than the previous case as the bondholders are taking more risk. This case is treated in Realdon (2004) but we do not consider it because, as explained in the latter, the issuance of exchangeable bonds when the issuer does not own the underlying shares is discouraged. This supports the affirmation that the issuer usually owns the shares and issues exchangeable bonds as a divesting strategy to dispose of the underlying shares in his possession.

3.3 Dynamic programming

Let $\mathscr{G}$ be a set of grid points $\{(a_1,b_1),(a_1,b_2),\ldots,(a_p,b_q)\}$ such that $\max(\Delta a_k,\Delta b_l) \to 0$ and $\mathbb{Q}[(V_t,r_t)\in[a_p,\infty)\times\mathbb{R}_+^*\cup\mathbb{R}_+^*\times[b_q,\infty)]\to 0$, when p and $q\to\infty$. Let $a_0=b_0=0$ and $a_{p+1}=b_{q+1}=\infty$. The rectangle $[a_i,a_{i+1})\times[b_j,b_{j+1})$ is designated by R_{ij}. Let $\Delta t=t_{n+1}-t_n$ a constant.

Dynamic programming acts as follows:

1. At date $t_N=T$, the value functions are known in closed form and are computed as described in Section 3.2.

2. At each date t_n, suppose that an approximation of each value function is available at a future decision date t_{n+1} on $\mathscr{G}$, indicated by $\tilde{f}_{n+1}(a_k,b_l)$, for $k=1,\ldots,p$ and $l=1,\ldots,q$, where f_n represents TB_n, BC_n, $D_n^{\bar{e}}$, D_n^e, or $\mathscr{E}_n$. Use a bilinear piecewise polynomial and interpolate each value function $\tilde{f}_{n+1}$ from $\mathscr{G}$ to the overall state space $[0,\infty)^2$ by setting:

$$\hat{f}_{n+1}(x,y)=\sum_{i=0}^{p}\sum_{j=0}^{q}\left(\alpha_{ij}^{n+1}+\beta_{ij}^{n+1}x+\gamma_{ij}^{n+1}y+\delta_{ij}^{n+1}xy\right)\mathbb{I}\left((x,y)\in R_{ij}\right),$$

   where the local coefficients of each value function f_{n+1}, α_{ij}^{n+1}, β_{ij}^{n+1}, γ_{ij}^{n+1}, and δ_{ij}^{n+1}, for $i=0,\ldots,p$ and $j=0,\ldots,q$, are the coefficients of the bilinear interpolation.

3. Approximate the expectation of every value function at t_n on $\mathscr{G}$:

$$\begin{aligned}&=\mathbb{E}\left[e^{-r\Delta t}\hat{f}_{n+1}(V_{t_{n+1}},S_{t_{n+1}})\mid(V_{t_n},S_{t_n})=(a_k,b_l)\right]\\&=e^{-r\Delta t}\sum_{i,j}\left(\alpha_{ij}^{n+1}T_{klij}^{00}+\beta_{ij}^{n+1}T_{klij}^{10}+\gamma_{ij}^{n+1}T_{klij}^{01}+\delta_{ij}^{n+1}T_{klij}^{11}\right),\end{aligned}$$

   where the transition tables T^{00},T^{10},T^{01}, and T^{11} are defined as follows:

$$\begin{aligned}T_{klij}^{\nu\mu} &= \mathbb{E}\Big[(V_{t_{n+1}})^{\nu}(S_{t_{n+1}})^{\mu}\mathbb{I}\left((V_{t_{n+1}},S_{t_{n+1}})\in R_{ij}\right)\mid\\ &\qquad (V_{t_n},S_{t_n})=(a_k,b_l)\Big],\quad \text{for } \nu \text{ and } \mu\in\{0,1\}.\end{aligned}$$

For example, T^{00}_{klij} represents the transition probability that the Markov process (V,S) moves from (a_k,b_l) at t_n and visits the rectangle R_{ij} at t_{n+1}. Closed-form solutions for these transition tables are given in Appendix 3.A.

4. Compute the value functions at t_n on $\mathscr{G}$ as described in Section 3.2.
5. Go to step 2 and repeat until $n = 0$.

This procedure is time consuming as we have a two-dimensional problem. Therefore, it makes sense to try to use parallel computing to accelerate the procedure. We parallelize our dynamic program by submitting the computation tasks associated to a given number of grid points to each available CPU. The algorithm used to parallelize our dynamic program is described in details in Appendix 3.B. This approach allows us to drastically reduce the computation time to a reasonable level.

We use the supercomputer Briarée managed by Calcul Québec and Compute Canada[4]. The code lines are written in C and compiled with GCC. We use the MPI library to access parallel computing.

3.4 Numerical investigation

In this section, we examine some characteristics of the exchangeable debt. For comparison purposes, we consider an exchangeable bond with the same parameters as in Realdon (2004). Considering an ordinary debt and an exchangeable bond both with a 5 year maturity and principal amount $P = 1$. The annual coupon rate is 3% for the ordinary debt and 4.7% for the exchangeable bond. Our numerical investigation presented here are based on a grid size of 300^2. A price calculation takes in average two minutes using parallel computing.

Figure 3.1 plots the value of the exchangeable bond as a function of the initial shares' level S_0 and the initial assets' value V_0. The exchangeable bond value is an increasing function of both variables. In fact, high values for the firm's assets represent less risky firms making the exchangeable bond more valuable. We also

4. The operation of this supercomputer is funded by the Canada Foundation for Innovation (CFI), Ministère de l'Économie, de la Science et de l'Innovation du Québec (MESI) and the Fonds de recherche du Québec - Nature et technologies (FRQ-NT).

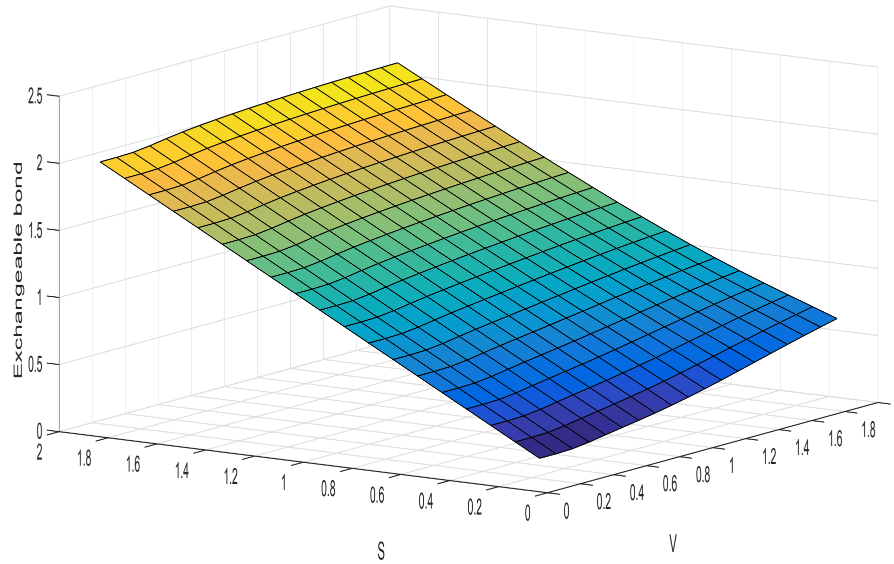

Figure 3.1: Exchangeable debt value as a function of the shares' value and the firm's assets value. The parameters used are $r = 0.04, \delta_1 = 0.3, \sigma_V = 02, \delta_2 = 0, \sigma_S = 0.3, \rho = 0, w = 0.2$, and $r^c = 0.35$.

notice that the increase in the shares' value has more significant impact on the exchangeable bond value than the increase in the firm's assets value. In fact, as S_0 increases, the bondholders are more likely to exercise the exchange option and the exchangeable bond value increases.

Figure 3.2 presents the exchangeable bond value as a function of the initial shares' value S_0 as the volatility of the firm's assets σ_V is changed. The exchangeable bond value decreases when the latter increases since the firm is more risky. For low values of the shares, the exchangeable bond value is more sensitive to changes in the firm risk-level; when the shares' value is low, the exchange option is less likely to be used and the bondholders are more exposed to the issuer risk.

Figure 3.3 illustrates that the exchangeable bond increases in the shares' volatility σ_S as the exchange option becomes more valuable. Figure 3.4 shows that the exchangeable bond rises when correlation rises. In fact, as explained by Realdon (2004), for negative values of ρ, high values of the issuer's assets are most likely associated to low values of the underlying shares, and vice versa. This situation is most valuable for the exchangeable bond. As correlation becomes positive, high

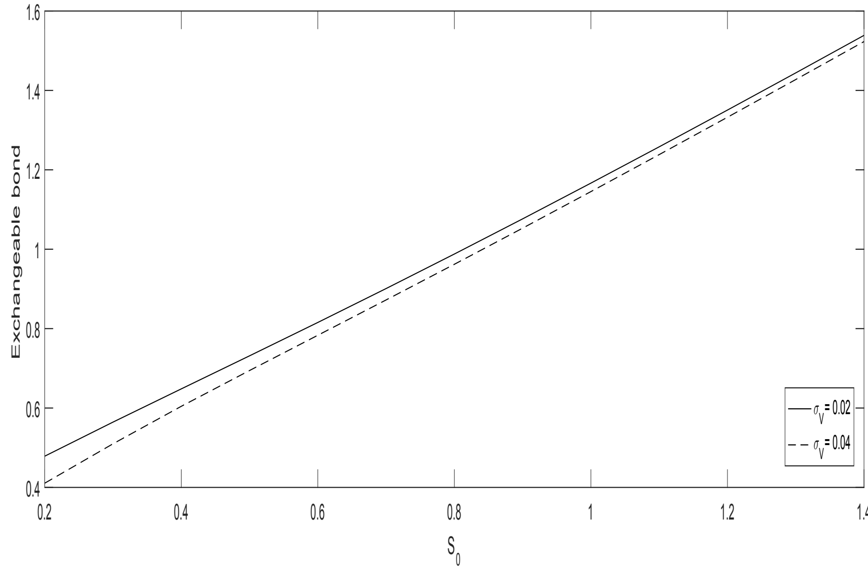

Figure 3.2: Exchangeable debt value as a function of the shares' value as the assets's volatility is changed. The parameters used are $r = 0.04, V_0 = 1.5, \delta_1 = 0.3, \delta_2 = 0, \sigma_s = 0.3, \rho = 0, w = 0.2$, and $r^c = 0.35$.

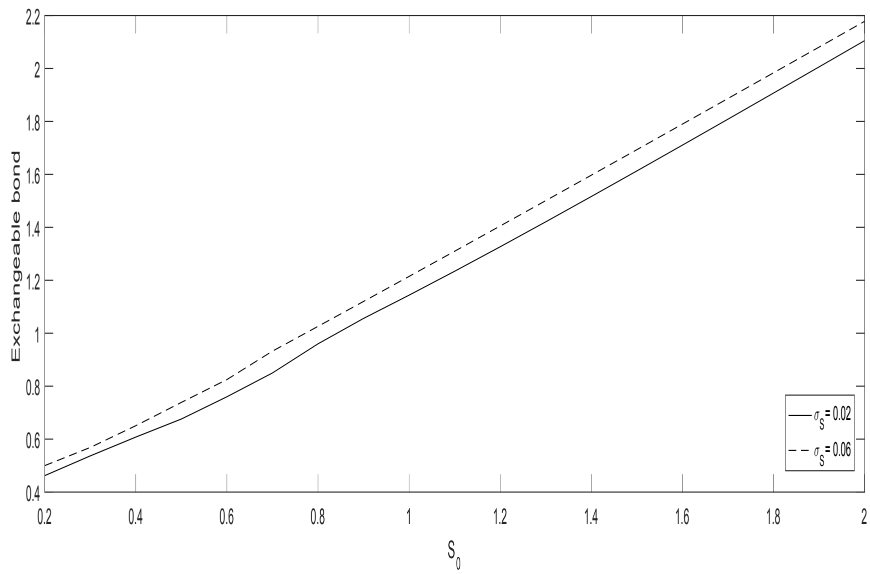

Figure 3.3: Exchangeable debt value as a function of the shares' value as the shares' volatility is changed. The parameters used are $r = 0.04, V_0 = 1.5, \delta_1 = 0.3, \sigma_V = 0.2, \delta_2 = 0, \rho = 0, w = 0.2$, and $r^c = 0.35$.

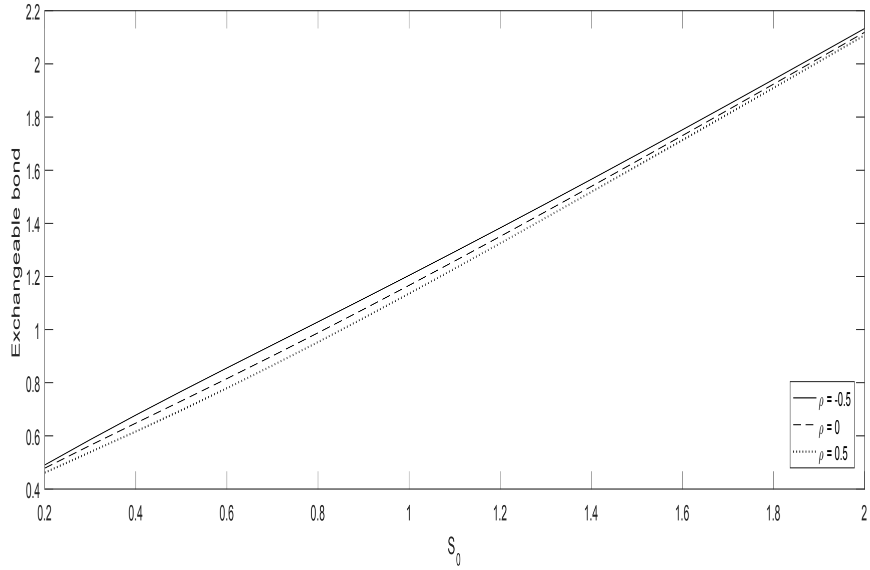

Figure 3.4: Exchangeable debt value as a function of the shares' value as the correlation is changed. The parameters used are $r = 0.04, \sigma_V = 0.2, \delta 1 = 0.3, V_0 = 1.5, \sigma_S = 0.3, \delta_2 = 0, w = 0.2$, and $r^c = 0.35$.

values of the issuer's assets are most likely associated with high values of the underlying shares, and vice versa. The first scenario is beneficial to the bondholders, but the second drives down the exchangeable bond value. Besides, increasing the the proportional bankruptcy costs and the nominal amount of the issuer's other outstanding debt decrease the value of the exchangeable bond as it increases the the loss given default.

3.5 Conclusion

In this paper, we propose a valuation framework for a hybrid-form of convertible debt, namely the exchangeable bond. This structured contract continues to gain popularity in corporate finance but is still less studied in terms of valuation purposes. Hence, we propose a general structural model for valuing exchangeable bonds in a setting that accounts for flexible debt structures, presence of bankruptcy costs and tax benefits. The model is solved using two-dimensional dynamic programming coupled with finite elements and parallel computing. The relevance of our methodology is that it can accommodate other styles of two-dimensional structured financial contracts such as reverse convertible bonds.

APPENDIX

3.A Transition parameters

The transition parameters $T_{klij}^{\nu\mu}$ for ν and $\mu \in \{0,1\}$, $k \in \{1,\ldots,p\}$, $l \in \{1,\ldots,q\}$, $i \in \{0,\ldots,p\}$, and $j \in \{0,\ldots,q\}$ are calculated as follows:

$$\begin{aligned}
T_{klij}^{00} &= \mathbb{E}^*\left[\mathbb{I}\left((V_{t_{n+1}},S_{t_{n+1}}) \in R_{ij}\right) \mid (V_{t_n},S_{t_n}) = (a_k,b_l)\right] \\
&= \mathbb{Q}\left[(V_{t_{n+1}},S_{t_{n+1}}) \in R_{ij} \mid (V_{t_n},S_{t_n}) = (a_k,b_l)\right] \\
&= \int_{x_{k,i}}^{x_{k,i+1}} \int_{y_{l,j}}^{y_{l,j+1}} \phi(z_1,z_2,\rho) dz_1 dz_2 \\
&= \Phi(x_{k,i+1},y_{l,j+1},\rho) - \Phi(x_{k,i},y_{l,j+1},\rho) - \Phi(x_{k,i+1},y_{l,j},\rho) + \Phi(x_{k,i},y_{l,j},\rho),
\end{aligned}$$

where

$$\begin{aligned}
x_{k,i} &= \left(\log\left(a_i/a_k\right) - \left(r - d_1 - \sigma_V^2/2\right)\Delta t\right) / \left(\sigma_V \sqrt{\Delta t}\right) \\
y_{l,j} &= \left(\log\left(b_j/b_l\right) - \left(r - d_2 - \sigma_S^2/2\right)\Delta t\right) / \left(\sigma_S \sqrt{\Delta t}\right).
\end{aligned}$$

The functions $\phi(\cdot,\cdot,\rho)$ and $\Phi(\cdot,\cdot,\rho)$ are respectively the density and the cumulative density functions of the bivariate standard normal distribution with correlation coefficient ρ. The function $\Phi(\cdot,\cdot,\rho)$ is computed according to Genz (2004).

$$
\begin{aligned}
T^{10}_{klij} &= \mathbb{E}^*\left[V_{t_{n+1}}\mathbb{I}\left((V_{t_{n+1}},S_{t_{n+1}})\in R_{ij}\right) \mid (V_{t_n},S_{t_n})=(a_k,b_l)\right]\\
&= \int_{x_{k,i}}^{x_{k,i+1}}\int_{y_{l,j}}^{y_{l,j+1}} a_k\exp\left((r-d_1-\sigma_V^2/2)\Delta t+\sigma_V\sqrt{\Delta t}z_1\right)\phi(z_1,z_2,\rho)dz_1dz_2\\
&= w_k^1\int_{x_{k,i}-\sigma_V\sqrt{\Delta t}}^{x_{k,i+1}-\sigma_V\sqrt{\Delta t}}\int_{y_{l,j}-\rho\sigma_V\sqrt{\Delta t}}^{y_{l,j+1}-\rho\sigma_V\sqrt{\Delta t}}\phi(u_1,u_2,\rho)du_1du_2\\
&= w_k^1\Big[\Phi(x_{k,i+1}-\sigma_V\sqrt{\Delta t},y_{l,j+1}-\rho\sigma_V\sqrt{\Delta t},\rho)-\\
&\quad\Phi(x_{k,i}-\sigma_V\sqrt{\Delta t},y_{l,j+1}-\rho\sigma_V\sqrt{\Delta t},\rho)-\\
&\quad\Phi(x_{k,i+1}-\sigma_V\sqrt{\Delta t},y_{l,j}-\rho\sigma_V\sqrt{\Delta t},\rho)+\\
&\quad\Phi(x_{k,i}-\sigma_V\sqrt{\Delta t},y_{l,j}-\rho\sigma_V\sqrt{\Delta t},\rho)\Big],
\end{aligned}
$$

where $w_k^1 = a_k\exp\left((r-d_1-\sigma_V^2/2)\Delta t+\sigma_V^2\Delta t/2\right)$.

$$
\begin{aligned}
T^{01}_{klij} &= \mathbb{E}^*\left[S_{t_{n+1}}\mathbb{I}\left((S_{t_{n+1}},S_{t_{n+1}})\in R_{ij}\right) \mid (S_{t_n},S_{t_n})=(a_k,b_l)\right]\\
&= \int_{x_{k,i}}^{x_{k,i+1}}\int_{y_{l,j}}^{y_{l,j+1}} b_l\exp\left((r-d_2-\sigma_S^2/2)\Delta t+\sigma_S\sqrt{\Delta t}z_2\right)\phi(z_1,z_2,\rho)dz_1dz_2\\
&= w_l^2\int_{x_{k,i}-\rho\sigma_S\sqrt{\Delta t}}^{x_{k,i+1}-\rho\sigma_S\sqrt{\Delta t}}\int_{y_{l,j}-\sigma_S\sqrt{\Delta t}}^{y_{l,j+1}-\sigma_S\sqrt{\Delta t}}\phi(u_1,u_2,\rho)du_1du_2\\
&= w_l^2\Big[\Phi(x_{k,i+1}-\rho\sigma_S\sqrt{\Delta t},y_{l,j+1}-\sigma_S\sqrt{\Delta t},\rho)-\\
&\quad\Phi(x_{k,i}-\rho\sigma_S\sqrt{\Delta t},y_{l,j+1}-\sigma_S\sqrt{\Delta t},\rho)-\\
&\quad\Phi(x_{k,i+1}-\rho\sigma_S\sqrt{\Delta t},y_{l,j}-\sigma_S\sqrt{\Delta t},\rho)+\\
&\quad\Phi(x_{k,i}-\rho\sigma_S\sqrt{\Delta t},y_{l,j}-\sigma_S\sqrt{\Delta t},\rho)\Big],
\end{aligned}
$$

where $w_l^2 = b_l \exp\Big((r-d_2-\sigma_S^2/2)\Delta t + \sigma_S^2 \Delta t/2\Big)$.

$$\begin{aligned}
T_{klij}^{11} &= \mathbb{E}^*\Big[V_{t_{n+1}} S_{t_{n+1}} \mathbb{I}\big((V_{t_{n+1}}, S_{t_{n+1}}) \in R_{ij}\big) \mid (V_{t_n}, S_{t_n}) = (a_k, b_l)\Big] \\
&= \int_{x_{k,i}}^{x_{k,i+1}} \int_{y_{l,j}}^{y_{l,j+1}} a_k \exp\Big((r-d_1-\sigma_V^2/2)\Delta t + \sigma_V \sqrt{\Delta t} z_1\Big) \times \\
&\quad b_l \exp\Big((r-d_2-\sigma_S^2/2)\Delta t + \sigma_S \sqrt{\Delta t} z_2\Big) \phi(z_1, z_2, \rho) dz_1 dz_2 \\
&= w_k^1 w_l^2 \exp(\rho \sigma_V \sigma_S \Delta t) \times \\
&\quad \int_{x_{k,i}-(\sigma_V+\rho\sigma_S)\sqrt{\Delta t}}^{x_{k,i+1}-(\sigma_V+\rho\sigma_S)\sqrt{\Delta t}} \int_{y_{l,j}-(\rho\sigma_V+\sigma_S)\sqrt{\Delta t}}^{y_{l,j+1}-(\rho\sigma_V+\sigma_S)\sqrt{\Delta t}} \phi(u_1, u_2, \rho) du_1 du_2 \\
&= w_k^1 w_l^2 \exp(\rho \sigma_V \sigma_S \Delta t) \times \\
&\quad \Big[\Phi(x_{k,i+1} - (\sigma_V + \rho\sigma_S)\sqrt{\Delta t}, y_{l,j+1} - (\rho\sigma_V + \sigma_S)\sqrt{\Delta t}, \rho) - \\
&\quad \Phi(x_{k,i} - (\sigma_V + \rho\sigma_S)\sqrt{\Delta t}, y_{l,j+1} - (\rho\sigma_V + \sigma_S)\sqrt{\Delta t}, \rho) - \\
&\quad \Phi(x_{k,i+1} - (\sigma_V + \rho\sigma_S)\sqrt{\Delta t}, y_{l,j} - (\rho\sigma_V + \sigma_S)\sqrt{\Delta t}, \rho) + \\
&\quad \Phi(x_{k,i} - (\sigma_V + \rho\sigma_S)\sqrt{\Delta t}, y_{l,j} - (\rho\sigma_V + \sigma_S)\sqrt{\Delta t}, \rho)\Big].
\end{aligned}$$

3.B Parallel computing algorithm

Parallel computing uses multiple central processing units (CPUs) simultaneously to speed-up complex computations. The Message Passing Interface (MPI) library allows the computing process to exchange information between the running CPU environments in order to achieve a given job. Each CPU has access to a certain memory space. MPI requires case-sensitive programming changes from the serial code to its parallel version.

The easiest way to parallelize DP is to submit the computation tasks associated to a given grid point (a_k, b_l), for $k = 1, \ldots, p$ and $l = 1, \ldots, q$, to a single CPU. Our parallel code acts as follows.

1. This single CPU computes once and locally stores the overall grid points

(a_i, b_j) and each value function values $f_N(a_i, b_j)$, for $i = 1, \ldots, p$ and $j = 1, \ldots, q$.

2. It also computes once and locally stores the $4 \times (p+1)(q+1)$ transition parameters T^{00}_{klij}, T^{10}_{klij}, T^{01}_{klij}, and T^{11}_{klij}, for $i = 0, \ldots, p$ and $j = 0, \ldots, q$.
3. It computes and stores at step $n+1$ the local coefficients α^{n+1}_{ij}, β^{n+1}_{ij}, γ^{n+1}_{ij}, and δ^{n+1}_{ij}, for each value function f_{n+1}, for $i = 0, \ldots, p$ and $j = 0, \ldots, q$.
4. It computes and stores at step n every value function $\tilde{f}_n(a_k, b_l)$.
5. The same CPU exports $\tilde{f}_n(a_k, b_l)$ to a selected CPU, the so-called master CPU.
6. The master CPU collects $\tilde{f}_n(a_k, b_l)$, for $k = 1, \ldots, p$ and $l = 1, \ldots, q$, and sends them back to all running CPUs.
7. Go to step 3 and repeat until $n = 0$.

Since the number of CPUs available to the analyst is usually less than the grid size pq, we submit the same number of grid points to each CPU.

BIBLIOGRAPHY

Ayadi, M. A., Ben-Ameur, H., and Fakhfakh, T. (2016). A dynamic program for valuing corporate securities. *European Journal of Operational Research*, 249(2):751–770.

Barber, B. M. (1993). Exchangeable debt. *Financial Management*, 22(2):48–60.

Brennan, M. J. and Schwartz, E. S. (1977). Convertible bonds: Valuation and optimal strategies for call and conversion. *The Journal of Finance*, 32(5):1699–1715.

Genz, A. (2004). Numerical computation of rectangular bivariate and trivariate normal and t probabilities. *Statistics and Computing*, 14(3):251–260.

Grimwood, R. and Hodges, S. (2002). The valuation of convertible bonds: a study of alternative pricing models. Working paper, Warwick Finance Research Institute.

Guo, B. and Ren, R. (2009). Pricing exchangeable bonds based on monte carlo method. In *2009 International Conference on Management and Service Science*.

Holthausen, R. W., Leftwich, R. W., and Mayers, D. (1987). The effect of large block transactions on security prices: a cross-sectional analysis. *Journal of Financial Economics*, 19(2):237–267.

Ingersoll, J. E. (1977). A contingent-claims valuation of convertible securities. *Journal of Financial Economics*, 4(3):289–321.

Jones, E. and Mason, S. (1986). Equity-linked debt. *Midland Corporate Finance Journal*, pages 47–58.

Longstaff, F. A. and Schwartz, E. S. (2001). Valuing American options by simulation: a simple least-squares approach. *The Review of Financial Studies*, 14(1):113–147.

Mikkelson, W. H. and Partch, M. M. (1985). Stock price effects and costs of secondary distributions. *Journal of Financial Economics*, 14(2):165–194.

Realdon, M. (2004). Valuation of exchangeable convertible bonds. *International Journal of Theoretical and Applied Finance*, 7(06):701–721.

CONCLUSION

In this thesis, we propose a two-dimensional dynamic program coupled with finite elements and parallel computing for valuing two-dimensional financial derivatives. In the first essay we present a model for valuing two-dimensional American-style options. In the second essay, we propose a structural model for valuing risky debt that takes into account both default risk and interest rate risk. Finally, in the third essay, we present a valuation framework for exchangeable bond within a structural model.

Our methodology, based on dynamic programming, presents two major advantages with respect to its competitors, given that it assumes a space but not a time discretization, and a numerical but not a statistical error. Our investigation shows that dynamic programming competes well against its alternative methodologies in terms of accuracy.

Future research avenues include extending our dynamic program to more complex state processes, such as two-dimensional jump diffusions and GARCH processes, as long as the transition parameters can be computed efficiently. Dynamic programming can also be extended to higher dimensions, which is challenging but feasible. We will still be using parallel computing, but dynamic programming can firstly be combined with quasi-Monte Carlo in moderate state-space dimensions. Applications in this context can be valuing multi-dimensional derivatives as well as adding additional factors to our credit risk model.