FORWARD AND INVERSE AMERICAN OPTION PRICING VIA A COMPLEMENTARITY APPROACH

BY

FUYUAN WANG

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Industrial Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Doctoral Committee:

Associate Professor Liming Feng, Chair, Co-Director of Research
Professor Jong-Shi Pang, University of Southern California, Co-Director of Research
Assistant Professor Jackie Shen
Associate Professor Yanfeng Ouyang

Abstract

This dissertation considers three topics. The first part discusses the pricing of American options under a local volatility model and two jump diffusion models: Kou's jump diffusion model and the Dupire system. In Chapter 2, we establish partial differential complementarity systems for pricing American options under the aforementioned three models. We also introduce two different discretization schemes, a finite difference method and a finite element method, for the discretization of the complementarity systems into a collection of linear complementarity problems (LCPs). In Chapter 3, we discuss four popular existing numerical algorithms—a PSOR method, a two phase active-set method, a semi-smooth Newton method and a pivoting method—for solving LCPs that arise under Kou's jump diffusion model and the Dupire system. The numerical results presented in the thesis summarize the effectiveness of each approach for solving the corresponding LCPs.

In the second part, we consider the calibration problems of computing an implied volatility parameter for American options under the Dupire system and the local volatility model. In Chapter 4, we formulate the calibration problem as an inverse problem of the forward pricing problem, which is modeled as a discretized partial differential linear complementarity system in Chapter 2. The resulting inverse problem then becomes an instance of a mathematical program with complementarity constraints (MPCC). Two methods for solving MPCCs, an implicit programming algorithm (IMPA) and a new hybrid algorithm, are studied in this dissertation. We test both algorithms and report their numerical performance for solving MPCCs derived under the Dupire system and the local volatility model with synthetic and market data.

In the third part of this thesis, we investigate a new class of MPCCs, a doubly uni-parametric MPCC, for which the calibration of American options under the Black-Scholes-Merton (BSM) model is a special case. In particular, we consider one new algorithm for solving this problem when the problem matrices are positive definite, and a second algorithm for the more general case when the matrices are merely positive semi-definite. We study the convergence of both algorithms based on the local stability of the solutions as well as the numerical performance of both algorithms for solving doubly uni-parametric MPCCs with tridiagonal matrices, which are applicable for the calibration problems under the BSM model.

To my parents,

for your constant love and support.

Acknowledgments

This thesis would not have been possible without the help of many people. In particular, I would like to give my special thanks to:

My parents - No words can possibly describe my deep gratitude towards my farther Jianguo Wang and my mother, Ning Liu. Without your tireless love and numerous sacrifices over the years, I would never have made it so far.

My supervisors: Jong-Shi Pang and Liming Feng - I would like to express my great thankfulness to have you as my mentors. Besides your financial support over the years which makes the completion of my PhD study possible, your patience and guidance to me is instrumental. Not only you have been a great source of knowledge and inspiration, but also your sound advises about writing, working and life have been and will be benefiting me throughout my life. For this and so much more I am sincerely grateful.

My friends: Alok Tiwari, Dane Schiro, Daniel Robinson, John Jossey, Yu-Ching Lee - Alok, thank you for being a great friend and always willing to help me in every possible ways. John, thank you for organizing the study group and helping me learning new skills. Dane and Yu-Ching, thank you for your help with the thesis. Daniel, I am grateful to have you by my side. Your incredible support, patience and encouragement mean so much to me.

My committee members: Jackie Shen and Yanfeng Ouyang - I would like to thank you for being on my committee and your time, efforts and valuable suggestions.

My roommates: Sibo Liu and Xiaoran Liu - Thank you for being so encouraging and considerate. Life would not been so cheerful without your company.

ISE staff: Amy summers, Debra Hilligoss, Donna Eiskamp, Holly Kizer, Lynnell Lacy and others - I am grateful to such a wonderful line of staff at this department. I especially want to thank Amy for her continuous help arranging the meetings with Prof. Pang over the years, and Holly for guiding me through the whole process.

Table of Contents

List of Tables

List of Symbols

Space

$\mathbf{R}^n$	the real n-dimensional space
$\mathbf{R}^n_+$	the non-negative orthant of $\mathbf{R}^n$
$\mathbf{R}^{n\times m}$	the space of $n \times m$ real matrices

Matrices

M	$\equiv (M_{ij})$; a matrix with entries M_{ij}
M^T	the transpose of a matrix M
$\bar{M}$	the comparison matrix of M with entries $\bar{M}_{ii} = \lvert M_{ii}\rvert$ and $\bar{M}_{ij} = -\lvert M_{ij}\rvert$, $j \neq i$ for all $i = 1, 2, \ldots, n$
M_{sym}	$\equiv \frac{1}{2}\left(M + M^T\right)$; the symmetric part of a matrix M
M_{skew}	$\equiv \frac{1}{2}\left(M - M^T\right)$; the skew symmetric part of a matrix M
$\lambda_{\max}(M)$	the largest eigenvalue of a symmetric positive definite matrix M
$\lambda_{\min}(M)$	the smallest eigenvalue of a symmetric positive definite matrix M
$M_{i,\cdot}$	the i-th row of M
$M_{\cdot,\alpha}$	the columns of M indexed by α
$M_{\alpha\beta}$	the sub-matrix of M with rows and columns indexed by α and β, respectively

Scalars

$\mathbf{R}$	the set of all real numbers
$\mathbf{R}^+$	the non-negative real numbers
t^+	$\equiv \max(0, t)$; the non-negative part of a scalar

Vectors

x^T	$\equiv (x_1, x_2, \ldots, x_n)$; the transpose of a vector x with components x_i
x_α	sub-vector of x with components indexed by α

$\{x_k\}$	a sequence of vectors $x_1, x_2, x_3, \ldots$
$x^T y$	the standard inner product of vectors in $\mathbf{R}^n$
$\|x\|$	the l_2-norm of a vector $x \in \mathbf{R}^n$, unless otherwise specified
$\|x\|_\infty$	$\equiv \max_{1 \le i \le n} \lvert x_i \rvert$; the l_∞-norm of a vector $x \in \mathbf{R}^n$
$x \geq y$	the (usual) partial ordering: $x_i \geq y_i$, $i = 1, 2, \ldots, n$
$x > y$	the strict ordering: $x_i > y_i$, $i = 1, 2, \ldots, n$
$\min(x, y)$	the vector whose i-th component is $\min(x_i, y_i)$
$dist(x, y)$	the Euclidean distance between x and y
$x \perp y$	x and y are perpendicular
e^i	the i-th coordinate vector whose length is determined by the context in which it is used

Functions

$F : \mathcal{D} \to \mathcal{R}$	a mapping with domain $\mathcal{D}$ and range $\mathcal{R}$
JF	$\equiv \left(\frac{\partial F_i}{\partial x_j}\right)$; the $m \times n$ Jacobian of a mapping $F : \mathbf{R}^n \to \mathbf{R}^m$ $(m \geq 2)$
$\nabla\theta$	$\equiv \left(\frac{\partial \theta}{\partial x_j}\right)$; the gradient of a function θ: $\mathbf{R}^n \to \mathbf{R}$
Jac F	the limiting Jacobian of $F : \mathbf{R}^n \to \mathbf{R}^m$
∂F	conv Jac F; the Clarke Jacobian of $F : \mathbf{R}^n \to \mathbf{R}^m$

Sets

$\in$, $\notin$	element membership, non-membership in a set
$\emptyset$, $\subseteq$, $\subset$	the empty set, set inclusion, proper set inclusion
$\cup$, $\cap$, $\times$	union, intersection, Cartesian product
ΠS_i	Cartesian product of sets S_i
$S_1 \setminus S_2$	the difference of two sets S_1 and S_2
∂S	the topological boundary of a set S
conv S	the convex hull of a set S

Problem classes and fundamental objects

$CP(F)$	complementarity problem defined by the function F
$LCP(q, M)$	linear complementarity problem defined by the vector q and the matrix M
$SOL(q, M)$	the solution set of the $LCP(q, M)$

Matrix classes

positive definite	matrices M such that $x^T Mx > 0$ for all $x \neq 0$
positive semi-definite	matrices M such that $x^T Mx \geq 0$ for all x
P	matrices M for which $\forall x \neq 0$, $\exists\, i$ such that $x_i(Mx)_i > 0$

CP functions

$\phi_{\mathrm{FB}}(a, b)$	$\equiv \sqrt{a^2 + b^2} - a - b$; the Fischer-Burmeister C-function
$\mathbf{F}_{\mathrm{FB}}(x)$	$\equiv (\phi_{\mathrm{FB}}(x_i, F(x_i)))_{i=1}^n$; the reformulation function of the $CP(F)$ for the Fischer-Burmeister C-function
θ_{FB}	$\equiv \frac{1}{2}\sum_{i=1}^n \phi_{\mathrm{FB}}^2(x_i, F(x_i))$; the merit function induced by the Fischer-Burmeister C-function

Chapter 1

Introduction

Option contracts have been known for many decades, and are among the most actively traded securities in the derivative market. Their prices are determined by other basic underlying assets, including stocks, fixed income securities and commodities, etc. Options provide investors protection from unfavorable price changes but at the same time benefit from favorable movements. To be more specific, the buyer of an option gains the right, but not the obligation, to engage in a specific transaction by paying the premium. There are mainly two basic kinds of options, a call option which assigns the holder the right to buy a certain asset at a specific price and a put option which assigns the holder the right to sell a certain asset at a specific price. The specific price is called the strike price. The action of buying and selling the underlying asset is called exercising the option. Usually, each option has an expiration date. If the option is not exercised prior to the expiration date, it becomes worthless. Based on the specific time periods at which options can be exercised, we can categorize options into different types, European options and American options. European options can only be exercised at their respective expiration dates, while American options can be exercised at any time prior to their expiration dates. Accurate pricing of options is of vital importance for maintaining an efficient derivative market. However, because of the possibility of early exercise, in general it is more difficult to price American options than to price European options.

Various models have been proposed by researchers for the accurate pricing of options. Among them, the most well-known option pricing model is the Black-Scholes-Merton (BSM) model in [BS73] and [Mer73]. One advantage of the BSM model is that it provides a closed-form formula for pricing European options. Moreover, one can further hedge his position using such an option formula. However, BSM has its limitations due to the basic assumptions of the model. First, BSM assumes that the price of the underlying asset follows a geometric Brownian motion, which fails to capture the extreme movements observed in the market [MT97]. Second, BSM also assumes a fixed volatility for different strike prices, contradicting the volatility smile exhibited by the implied volatility [EG02].

Alternative classes of models have been studied to remedy the deficiencies of the BSM model. One popular class of models, the jump diffusion model employs a Levy process to capture the volatile behaviors, such as jumps observed in the market. Jump diffusion models have been extensively studied by many researchers, including Andersen and Andreasen in [AA00], Bates in [Bat96], Kou in [Kou02] and Merton in [Mer76]. Another popular class of models is proposed based on the modification of the constant volatility assumption in the BSM model. For example, the volatility is a function of the current underlying asset and time in the local volatility model. This concept was first proposed by Dupire in [Dup94], Derman and Kani in [DK94]. In a stochastic volatility model [HW87], the volatility is captured by another stochastic process. Bates in [Bat96] expanded this stochastic volatility model by introducing jumps in the asset return process. Later on, Duffie with his colleagues in [DPS00] further expanded the same model by allowing jumps in both the asset return process and the variance process.

Once we determine a specific model for the option evaluation, we also need to know the parameters of the specific model in order to price options accurately. As is known, these parameters are in general not constants. Instead, they are highly complicated functions of deterministic and random factors. Different approaches have been proposed to deal with the calibration problems of unknown parameters. Popular calibration methods include statistical estimation methods based on historical data (see [Gib91]) and a mathematical modeling of parameters (see [HW87], [JS87] and [ZFV98]). In addition to the above two types of calibration methods, another popular approach to calibrate these parameters is by solving an optimization problem formed from the observed values of the options. One way of utilizing the observed values is by setting the objective function of the optimization problem as the minimization of the deviation between theoretical prices and observed market prices. This idea was originally suggested by Rendleman in [LR76] and empirically tested by Beckers in [Bec81].

In the case of American option pricing, we know there is no explicit formula or finite procedure for computing the exact option prices. However, the price of an American option must satisfy a partial differential complementarity system. Upon a suitable discretization of the partial differential complementarity system, we obtain a finite-dimensional variational inequality, whose solution can be obtained by solving a linear complementarity problem (LCP). Therefore, fast solution methods for solving general complementarity problems and LCPs are of great importance to computational finance and also, one of the interests of this thesis work.

Traditional methods for solving complementarity problems and LCPs include semi-smooth Newton methods (see [HIK02] and [FP03a]) and pivoting methods (see [CPS09]). Other practical LCP solvers include a projected successive over-relaxation (PSOR) method and its variants, such as the projected Gauss-Seidel method [CPS09]. Recently, active-set methods have been used for solving LCPs with improved efficiency. For example, Borici and Luethi in [BL05] proposed a simplex-like method for solving LCPs with Z-matrices. In [IT04], Ikonen and Toivanen developed and tested four methods—a projected multi-grid method, an operator splitting method, a penalty method and a component-wise splitting method—for solving LCPs. Their numerical results further suggested that the component-wise splitting method had the best performance among the four methods. Recently, Feng, Linetsky, Morales and Norcedal in [FLMN10] used a two phase active-set method to solve LCPs arising from the pricing of American options. The two phase active-set method combines iterations of the PSOR method together with reduced-space steps. It yields very good numerical results, yet lacks a convergence proof. The convergence result of a variant of the two phase active-set method is obtained in the subsequent paper [RFNP11]. Other popular treatment of LCPs arising from American option pricing problems can be found in [HP98] and [WHD95]. In our study, we compare the numerical performance of four LCP solvers, a PSOR method, a two phase active-set method, a semi-smooth Newton method and a pivoting method for solving option pricing problems derived under different models to gain a thorough perspective of the respective performances.

In terms of the calibration of option pricing models, we are particularly interested in calibrating the unknown volatility parameter and adopt an optimization method to solve this class of calibration problems in the thesis. Specifically, we use a discrepancy measure of the unknown volatility as the objective function of the optimization problem. As mentioned in the previous paragraphs, the idea of using an optimization approach to deal with the uncertain volatility is not new and many calibration methods have been proposed for solving this class of optimization problems. In the case of European options, traditional methods used to solve the optimization problems derived from the calibration problems include employing a relative entropy minimization problem to calibrate the volatility function (see [Ave98]) and gradient descent methods (see [LO97a] and [LO97b]). Coleman, Li and Verma in [CLV99] suggested using a 2-D spline approximation coupled with a finite-dimensional constrained nonlinear optimization to reconstruct the smooth local volatility function for European options. In the case of American options, Achdou and his co-authors in [AIP$^+$04], [AP05] and [Ach08] considered calibrating local volatility models using a finite element method. A non-parametric calibration of the local volatility model has been proposed by Huang and Pang in [HP00], which closely reproduced the observed market prices of American options. In Huang-Pang's work, the calibration of the local volatility function was formulated as a mathematical program with complementarity constraints (MPCC) and solved by an implicit programming algorithm (IMPA). Expanding on Huang-Pang's work, we aim to develop a new calibration method that computes the two-dimensional local volatility function with improved efficiency. The fundamental idea of this new method is to use special basis functions to construct the volatility function. The utilization of the basis functions in the construction of the local volatility function greatly reduces the size of the corresponding MPCC. In addition, we propose a new hybrid algorithm, which combines both IMPA iterations together with a grid search method for solving the formulated MPCCs. By employing both IMPA and a grid search method, the new hybrid algorithm results in a potential acceleration of the convergence.

In this thesis, we also consider solving a special class of MPCCs, doubly uni-parametric MPCCs. This type of optimization problems can be used to calibrate the single volatility parameter in the BSM model. By studying the special structure of the constraints in a doubly uni-parametric MPCC, we are able to develop innovative computational techniques to solve the problem efficiently.

Overall, this thesis achieves three goals.

1. Conduct an extensive computational survey on the relative numerical performances of different solvers for solving LCPs arising from pricing American style options under jump diffusion models. In particular, we study the numerical performance of three popular iterative LCP solvers, including a projected successive over-relaxation (PSOR) method in [CPS09], a two phase active-set method in [FLMN10] and a semi-smooth Newton method in [FP03a] along with a parametric pivoting method in [PC85].

2. Calibrate the volatility parameter based on the observed option prices. We follow the same approach presented in [HP00] and solve the calibration problems of American options by reformulating them as MPCCs. In our implementation, the objective function is chosen to minimize the square deviation between the theoretical option prices and the observed market prices.

3. Inspired by the calibration problems of American options under the BSM model, we design efficient algorithms to solve a special class of MPCCs, doubly uni-parametric MPCCs. We also establish convergence results for our proposed algorithms under certain mild assumptions. Specifically, we prove the convergence results of the proposed algorithms in the case that the objective functions are globally Lipschitz and the associated matrices are positive definite/positive semi-definite.

The above tasks are challenging because of the following consideration:

1. We encounter the curse of dimensionality when solving MPCCs with constraints of large sizes. For example, the dimension of our calibration problems of American options depends on both the sizes of the time and space discretizations. Therefore, it can be quite computationally expensive to calibrate a volatility function with a fine discretization. We would like to design an effective technique to handle MPCCs with large complementarity constraints.

2. One existing approach to solve MPCCs is to apply a descent-type method. Such an approach was first discussed in [PHR91], later expanded in [LPR96] and used in [HP00] for solving calibration problems of American options under the local volatility model. The fundamental idea of the algorithm is based

on the fact that a directional derivative of the solution to the complementarity constraints is the solution to a mixed LCP of reduced size along any direction (see [CPS09]). However, there is no convergence results of the algorithm at this point. In particular, we would like to design an efficient algorithm with guaranteed convergence for solving a special type of MPCCs, doubly uni-parametric MPCCs. We also would like to extend the new algorithm to solve more general MPCCs where the constraints on the associated matrices of MPCCs are more relaxed. Hence, the nature of our problems necessitates the development of new theories and techniques. We accomplish the task by conducting comprehensive studies of the special structure of the complementarity constraints in the doubly uni-parametric MPCCs.

The organization of this thesis is presented as follows. We first introduce the dynamics of the financial market and different models that are used in this thesis the pricing of American options in Chapter 2. Those models include the classic Black-Scholes-Merton model [BS73], the local volatility model, Kou's jump diffusion model [Kou02] and the Dupire system [Dup94]. We then present two popular discretization schemes, the finite difference method and the finite element method for transforming the complementarity systems derived under the aforementioned models into LCPs. Next, we present four different algorithms for solving LCPs arising from the pricing of American options under different models in Chapter 3. Those include a projected successive over-relaxation (PSOR) method in [CPS09], a two phase active-set method which combines iterations of the PSOR method with reduced-space steps in [FLMN10], a semi-smooth Newton method in [FP03a] and a parametric pivoting method in [CPS09]. We also report their numerical performances for solving the corresponding LCPs derived under different jump diffusion models. In Chapter 4, we calibrate the volatility parameter of American options via solving its reformulation as an MPCC. In particular, we solve the corresponding MPCC by modifying the existing IMPA employed in [HP00]. This modification greatly enhances IMPA's capability of solving MPCCs with large constraints. We test the modified IMPA for solving calibration problems derived under both the Dupire system and the local volatility model with different discretizations. The numerical performance of the modified IMPA is also compared with that of a grid search method to achieve a better understanding of the modified IMPA. In the latter part of Chapter 4, we also propose a hybrid algorithm developed from IMPA to further improve the computational efficiency. This hybrid algorithm, combing both IMPA iterations along with a grid search method, has greatly accelerated the convergence under our experiments. In Chapter 5, we developed two innovative algorithms for solving a specific type of MPCCs, doubly uni-parametric MPCCs. The two algorithms are designed for solving doubly uni-parametric MPCCs with different associated matrices. One is developed for the case of positive definite matrices, while the other is developed for the case of positive semi-definite matrices. For each algorithm, we first present a detailed description of the method, followed by new theories related to the convergence result of the algorithm. We then test the two new algorithms along with a one-dimensional grid search method and IMPA for solving doubly uni-parametric MPCCs. We summarize the numerical results in the last section of Chapter 5.

Chapter 2

Linear complementarity problems arising in American option pricing

As mentioned earlier in the introduction, when the price of an underlying asset follows a diffusion process, like the geometric Brownian motion, an American option pricing problem can be reformulated into a partial differential complementarity problem. Once discretized in space and time, the complementarity system can be converted into a large LCP solvable via a sequence of small LCPs. In this chapter, we will be mainly focusing on the reformulation of the evaluation problems of American options. Specifically, in Section 2.1, we review of the dynamics of a financial market and some existing popular models, including Kou's jump diffusion model, the Dupire system and a local volatility model. In Section 2.2, we then introduce the partial differential complementarity systems derived under the aforementioned models for the pricing of American options. We also present a detailed description of two discretization schemes, a finite difference method and a finite element method. Both schemes are then used to transform the complementarity systems derived under different models into LCPs.

2.1 Option pricing models

In this section, we first introduce the basic dynamics of a financial market, followed by some popular models other than the BSM model. These include a general jump diffusion model, which captures the volatile movements of the underlying asset by the introduction of a jump part, and a local volatility model, which explains the volatility smiles observed from the equity market.

2.1.1 Background: a financial market

One of the most fundamental assumptions about the financial market is that a stock price follows a continuous-variable and continuous-time process known as a geometric Brownian motion. To be more specific, suppose we have a time variable t which takes values from a finite interval $[0, T]$, T being the maturity of an option on the stock, and a constant positive risk-free interest rate $r > 0$, then any risk-free asset D_t

and stock S_t can be described by the following stochastic differential equations:

$$\begin{aligned} dD_t &= rD_t t dt, \\ dS_t &= \mu S_t dt + \sigma S_t dB_t, \end{aligned} \tag{2.1}$$

with μ being a constant drift rate and $\sigma > 0$ being a constant volatility for all $t \in [0, T]$. Here B_t, $t \in [0, T]$, is a standard Brownian motion with a mean change of zero and a variance rate of 1.0 per year. This stochastic process was first used by botanist Robert Brown in 1827 to describe the motion of a particle that was subject to large numbers of small molecular shocks. The process B_t is characterized by the following three properties:

- $B_0 = 0$ almost surely.
- The function $t \to B_t$ is continuous almost surely.
- B_t has independent increments with $B_t - B_s \sim N(0, t - s)$ for any $t > s > 0$.

Notice that in the last property, by independency we mean that for any $0 \leq s_1 < t_1 \leq s_2 < t_2$, we have $B_{t_1} - B_{s_1}$ and $B_{t_2} - B_{s_2}$ are independent random variables. More details about the Brownian motion and the basic assumptions of the financial market can be found in [Hul09, Chapter 9].

2.1.2 Jump diffusion model

The classic option pricing model is the Black-Scholes-Merton model. As mentioned earlier in the introduction, one of the drawbacks of the BSM model lies in the fact that the model does not allow jumps in the asset prices under the continuous stock prices assumption. Therefore, one of the natural extension of the BSM model is to introduce jumps in the asset prices to capture dramatic movements in the market.

If at time t the stock price jumps, we use S_{t-} and S_t to denote the stock prices before and after the jump takes place in a jump diffusion model. Similar to the second stochastic differential equation of (2.1), we can write the equation, which governs the underlying asset price S_t under a jump diffusion model as follows.

$$\frac{dS_t}{S_{t-}} = \mu dt + \sigma dB_t + d\left(\sum_{i=1}^{N_t}(J_i - 1)\right), \tag{2.2}$$

which is equivalent to

$$S_t = S_0 \exp\left((\mu - \frac{1}{2}\sigma^2)t + \sigma B_t\right) \prod_{i=1}^{N_t} J_i.$$

Usually, we apply a log transformation to S_t for the sake of computation. Let $x_t = \log(S_t) - \log K$, it is easy to verify that x_t is ruled by the stochastic differential equation

$$dx_t = (\mu - \frac{1}{2}\sigma^2)dt + \sigma dB_t + d\widehat{J}_t,$$

where $\widehat{J}_t = \sum_{i=1}^{N_t} \log(J_i)$. Notice that we multiply the stock price by a random variable J_i at the i-th jump. In this thesis, we consider the case where $\widehat{J}_t = \sum_{i=1}^{N_t} Z_i$ is a compound Poisson process with rate λ and independent, identically distributed jump magnitude $\{Z\}$. Therefore, we have $J_i = J \equiv e^Z$. Moreover, we have

$$\mu = r - q + \lambda\left(1 - \mathbf{E}(e^Z)\right), \tag{2.3}$$

with q being the dividend of the underlying asset under the equivalent martingale measure. This is because the expected change of S_t in a small time interval is

$$\mu S_t \Delta t + \mathbf{E}(J-1)\lambda S_t \Delta t.$$

Under the equivalent martingale measure, we require the expected change to be $(r-q)S_t\Delta t$. Hence, condition (2.3) holds. The distribution of the jump magnitude Z varies among different models. We present two special cases of the jump diffusion models, the BSM and Kou's jump diffusion model as follows.

- When $J \equiv 0$, the jump diffusion model becomes the classic BSM model [BS73]

$$dS_t = \mu S_t + \sigma S_t dB_t. \tag{2.4}$$

- When the magnitude distribution Z is a double exponential distribution with density

$$p(z) = p\eta_1 e^{-\eta_1 z}\mathbf{1}_{\{z \geq 0\}} + (1-p)\eta_2 e^{\eta_2 z}\mathbf{1}_{\{z<0\}}, \tag{2.5}$$

then we obtain Kou's jump diffusion model [Kou02]. We again have the equivalent martingale measure imposing the following condition on the parameters of Kou's jump diffusion model:

$$\mu = r - q + \lambda\left((1-p)(\eta_2+1)^{-1} - p(\eta_1-1)^{-1}\right). \tag{2.6}$$

The jump diffusion model is mainly proposed to explain the skew effects of option prices observed in the market. However, it has its own limitations. The jumps tend to be averaged out over the long run in the jump diffusion model. Hence, it does not work particularly well with options having longer maturities due to its inablility to produce enough skewness for those options.

2.1.3 Local volatility model

Notice that in both the BSM model and a jump diffusion model (2.2), the volatility parameter is constant throughout the whole process. However in practice, we know that traders usually assume that the volatility is dependent upon the strike price, the price of the current underlying asset and the maturity of the option. Therefore, to relax the constant restriction on the volatility, a local volatility model is proposed. In a local volatility model, the constant volatility is replaced by a function of time and the underlying asset as $\sigma(S_t, t)$. Similar to the stochastic differential equation in the BSM model (2.4), the stock price S_t follows

$$dS_t = \mu S_t + \sigma(S_t, t) S_t dB_t \tag{2.7}$$

under a local volatility model. The local volatility model was first introduced in [DK94] and [Dup94]. The main purpose of calibrating such a model is to find a volatility function $\sigma(S_t, t)$ which produces option prices that closely match the observed market prices. This model not only explains the relationship between a volatility smile and the risk-neutral probability distribution taken by the stock price S_t, but also is used as a pricing tool for the evaluation of exotic options and other derivatives.

2.2 Linear complementarity systems of various models

In this section, we present the partial differential complementarity systems obtained under jump diffusion models and a local volatility model, as well as two popular discretization schemes for transforming the corresponding complementarily systems into LCPs. Before we present the complementarity systems derived under different models, we first introduce a few definitions which are frequently used in this work and highly related to our pricing problems, followed by the assumptions that we employ throughout this thesis.

For $0 \leq t \leq T$, we use V_t to denote the value of an option, which can be viewed as a continuous function of the asset price S_t and time t: $V_t = V(S_t, t)$. If an option is not exercised before maturity, then its value equals its payoff function at the expiry time. Let K be the strike price of an option. For a call, the payoff function $\Psi(S_t) = \max(S_t - K, 0)$ is the profit obtained by exercising the option immediately. Therefore, if $S_T > K$, then the holder will choose to exercise the option and get an instant profit $S_T - K$; otherwise, the option will not be exercised and become valueless. Hence, the value of a call option at the maturity can be written as

$$V(S_T, T) = \max(S_T - K, 0) = (S_T - K)^+.$$

Similarly, the payoff function for a put option is $\Psi(S_t) = \max(K - S_t, 0)$, and the value of a put option at the maturity can be written as

$$V(S_T, T) = \max(K - S_T, 0) = (K - S_T)^+. \tag{2.8}$$

In order to price options, assumptions about the financial market are required. Throughout this thesis, we assume that the following statements are true:

1. There are no transaction costs.

2. Borrowing and lending are possible at risk-free interest rates.

3. There is no arbitrage in the market.

Based on the above assumptions, especially the arbitrage-free argument, we can transform American option pricing problems into complementarity systems.

2.2.1 Complementarity systems

In this subsection, we mainly focus on presenting the complementarity systems formulated from American pricing problems under the Black-Scholes-Merton model, the local volatility model, Kou's jump diffusion model and its variant: the Dupire system. The essence of the formulation is based upon the arbitrage-free argument. In the aforementioned financial market, we construct a portfolio of an option and the underlying asset in such a way that the return of this portfolio is bounded above by the risk-free interest rate. By imposing this constraint, we can derive the corresponding complementarity systems under different models.

Complementarity system of the BSM model

Under the basic framework of the Black-Scholes-Merton model [BS73], we can construct a risk-free portfolio that consists of the option and the stock such that the return of the portfolio is deterministic for an European option. By using the equivalent martingale measure, we again obtain the following classic Black-Scholes partial differential equation that is satisfied by the value of an European option:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2(S,t)S^2\frac{\partial^2 V}{\partial S^2} + (r-q)S\frac{\partial V}{\partial S} - rV = 0. \tag{2.9}$$

However, condition (2.9) will no longer hold throughout the lifetime of an American option due to its early exercise feature. Instead, the arbitrage-free argument leads to the following complementarity system that must be satisfied by an American option under the BSM model. For $t \in [0, T]$ and $S \in [0, +\infty)$,

$$\begin{aligned}
&\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2(S,t)S^2\frac{\partial^2 V}{\partial S^2} + (r-q)S\frac{\partial V}{\partial S} - rV \leq 0, \\
&V \geq \Psi, \\
&\left(\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2(S,t)S^2\frac{\partial^2 V}{\partial S^2} + (r-q)S\frac{\partial V}{\partial S} - rV\right) \cdot (V - \Psi) = 0,
\end{aligned} \tag{2.10}$$

with terminal condition

$$V(S,T) = \Psi(S),\ S \in [0, +\infty).$$

The first two inequalities of system (2.10) describe the properties of a risk-free portfolio that consists of an American option and the stock. Due to the early exercise feature, the return of this portfolio is always smaller or equal to the risk-free interest rate. Moreover, since the owner of an American option can choose to exercise the option at any time before its maturity, the value of an American option must be greater or equal to its payoff value. Hence, the first two inequalities of (2.10) hold. In addition, for any given state (S_t, t), the owner of an American option can choose either to exercise or to keep the American option. If the option is exercised, then the value of the option is equal to its payoff value; else the option is not exercised, then it can be treated as an European option. Therefore, the option must satisfies condition (2.9). Hence, the complementarity system in (2.10) follows.

In general, the solution to the complementarity system (2.10) does not have an explicit form. Hence, we apply discretization schemes to solve the system numerically. One usual approach is to transform the complementarity system (2.10) into a large LCP and apply LCP solvers to solve the corresponding LCP. In this thesis, we adopt this approach and apply different existing solvers to solve complementarity systems of form (2.10). Discussion about the existence and uniqueness of a solution to system (2.10) can be found in [JLL90].

Complementarity system of a local volatility model

Notice that the only difference between the BSM model and a local volatility model is that the constant volatility σ of the BSM model is replaced by a volatility function $\sigma(S_t, t)$. By following the same arbitrage-free argument discussed above, we can derive the conditions that must be satisfied by $V(S_t, t)$ in the local volatility model. We present the complementarity system satisfied by $V(S_t, t)$ under the local volatility model as follows. For $t \in [0, T]$ and $S \in [0, +\infty)$,

$$\begin{aligned}
&\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2(S,t)S^2\frac{\partial^2 V}{\partial S^2} + (r-q)S\frac{\partial V}{\partial S} - rV \le 0, \\
&V \ge \Psi, \\
&\left(\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2(S,t)S^2\frac{\partial^2 V}{\partial S^2} + (r-q)S\frac{\partial V}{\partial S} - rV\right) \cdot (V - \Psi) = 0,
\end{aligned} \tag{2.11}$$

with terminal condition

$$V(S,T) = \Psi(S),\ S \in [0, +\infty).$$

Note that the only difference between (2.10) and (2.11) is that we replace σ by $\sigma(S,t)$.

Complementarity system of Kou's jump diffusion model

Now we consider the complementarity system derived under Kou's jump diffusion model. In this model, we assume the jump magnitudes $\{Z_i\}$ are independent random variables with a density function $p(z)$ given in (2.5). Therefore, the value of an American option satisfies the following partial integro-differential inequalities (see [Tan04] for a detailed derivation). For $t \in [0, T]$ and $S \in [0, +\infty)$,

$$\begin{aligned}
&\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2} + \mu S\frac{\partial V}{\partial S} + \int_{\mathbf{R}} \left(V(Se^z,t) - V(S,t)\right)\nu(z)dz - rV \le 0, \\
&V \ge \Psi, \\
&\left(\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2} + \mu S\frac{\partial V}{\partial S} + \int_{\mathbf{R}} (V(Se^z,t) - V(S,t))\nu(z)dz - rV\right) \cdot (V - \Psi) = 0,
\end{aligned} \tag{2.12}$$

with terminal condition

$$V(S,T) = \Psi(S),\ S \in [0, +\infty).$$

Here, $\nu(\cdot)$ represents the Levy density of the underlying process S_t. In Kou's jump diffusion model, it is calculated by

$$\nu(z) = \lambda p(z).$$

For the convenience of discretization, we apply a log transformation x_t to the stock price S_t: $S_t = Ke^{x_t}$, where K is the strike price of the American option. By applying the chain rule to complementarity system (2.12), we then obtain the complementarity system satisfied by V_t in terms of x_t. For simplicity, we only present the resulting complementarity system without the derivation. For $t \in [0,T]$ and $x \in (-\infty,+\infty)$,

$$\begin{aligned} &\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 \frac{\partial^2 V}{\partial x^2} + (\mu - \frac{1}{2}\sigma^2)\frac{\partial V}{\partial x} + \lambda \int_{\mathbf{R}} (V(x+z) - V(x))\, p(z)dz - rV \le 0, \\ &V \ge \Psi, \\ &\left(\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 \frac{\partial^2 V}{\partial x^2} + (\mu - \frac{1}{2}\sigma^2)\frac{\partial V}{\partial x} + \lambda \int_{\mathbf{R}} (V(x+z) - V(x))\, p(z)dz - rV\right) \cdot (V - \Psi) = 0, \end{aligned} \tag{2.13}$$

with terminal condition

$$V(x,T) = \Psi(Ke^x),\ x \in (-\infty,+\infty).$$

Complementarity system of the Dupire system

In system (2.12), the price of an American option is calculated as a function of the stock price S_t and the current time t. In addition, the American option price can also be expressed in terms of the strike price K and the option maturity T. We can derive the corresponding complementarity system satisfied by $V(K,T)$ from system (2.12) by the change of variables. The complementarity system satisfied by $V(K,T)$ is summarized as follows. For $T \in [0,\infty)$ and $K \in [0,\infty)$,

$$\begin{aligned} &\frac{\partial V}{\partial T} - \mathcal{L}V \ge 0, \\ &V - \Psi \ge 0, \\ &(V - \Psi)\left(\frac{\partial V}{\partial T} - \mathcal{L}V\right) = 0, \end{aligned} \tag{2.14}$$

with the initial condition

$$V(0,K) = \Psi(K),\ K \in [0,\infty).$$

Here, $\mathcal{L}V = \frac{\sigma^2 K^2}{2}\frac{\partial^2 V}{\partial K^2} - \mu K \frac{\partial V}{\partial K} - (r-\mu)V + \int_{\mathbf{R}} (e^z V(Ke^{-z},T) - V(K,T))\,\nu(z)dz$.

The complementarity system (2.14) was first proposed by Dupire in [Dup94]. Such a system can be coupled with a calibration problem to greatly reduce the computational efforts. Specifically, for a given stock price and time to maturity, system (2.14) computes prices of multiple American options with different strike prices. System (2.12) derived under Kou's jump diffusion model computes only the price of an American option with a given strike price. If in a calibration problem, we are given prices of multiple American options with different strike prices, then working under the Dupire system instead of Kou's jump diffusion model leads to solving an optimization problem with fewer complementarity constraints (see Chapter 4 for more discussions).

The derivation of the Dupire system (2.14) is quite straightforward and presented as follows. Without loss of generosity, we use an American put option as an example to illustrate this derivation. For an American option, we can write the option price as a function of the current time t, the asset price S, the strike price K and the option maturity T. As shown in [Shr04, Chapter 8], the American put option price is the solution to the optimal stopping problem

$$\begin{aligned} V(t,S,T,K) &= \sup_{T\geq\tau\geq t} \mathbf{E}_{t,0}[e^{-r(\tau-t)}(K-Se^{x_\tau})^+] \\ &= \sup_{T-t>\tau\geq 0} \mathbf{E}_{0,0}[e^{-r\tau}(K-Se^{x_\tau})^+]. \end{aligned} \tag{2.15}$$

Here, x_t is the process for the log return of the underlying asset and $\mathbf{E}_{t,x}$ is the expectation conditioning on $x_t = x$. Denote $g(t,y) \equiv \sup_{t\geq\tau\geq 0} \mathbf{E}_{0,0}[e^{-r\tau}(1-ye^{x_\tau})^+]$, then equation (2.15) is equivalent to

$$\begin{aligned} V(t,S,T,K) &= \sup_{T-t\geq\tau\geq 0} K\mathbf{E}_{0,0}[e^{-r\tau}(1-\frac{S}{K}e^{x_\tau})^+] \\ &= Kg\left(T-t, \frac{S}{K}\right). \end{aligned} \tag{2.16}$$

Given the above expression (2.16) of $V(t,S,T,K)$, it is easy to verify that the following equations hold:

$$\begin{aligned} \frac{\partial V}{\partial T} &= -\frac{\partial V}{\partial t}, \\ \frac{\partial V}{\partial K} &= \frac{V}{K} - \frac{S}{K}\frac{\partial V}{\partial S}, \\ \frac{\partial^2 V}{\partial K^2} &= \frac{S^2}{K^2}\frac{\partial^2 K}{\partial x^2}. \end{aligned} \tag{2.17}$$

By substituting (2.17) to (2.12), we obtain the complementarity system (2.14).

We again apply a log transformation to the strike price K in system (2.14) for the convenience of the numerical implementation. For simplicity, we replace the variable T in (2.14) by t, and define

$$\psi(x) \equiv \Psi(Se^x),\ \bar{u}(x,t) \equiv V(Se^x, t).$$

Then $\bar{u}(x,t)$ solves the following complementarity system. For $t \in [0,+\infty)$ and $x \in (-\infty,+\infty)$,

$$\begin{aligned} &\frac{\partial \bar{u}}{\partial t} - \mathcal{A}\bar{u} - \mathcal{B}\bar{u} \geq 0, \\ &\bar{u} - \psi \geq 0, \\ &\left(\frac{\partial \bar{u}}{\partial t} - \mathcal{A}\bar{u} - \mathcal{B}\bar{u}\right)(\bar{u}-\psi) = 0, \end{aligned} \tag{2.18}$$

where

$$\mathcal{A}\bar{u} = \frac{\sigma^2}{2}\frac{\partial^2 \bar{u}}{\partial x^2} - (\mu + \frac{\sigma^2}{2})\frac{\partial \bar{u}}{\partial x} - (r - \mu + \lambda)\bar{u},$$
$$\mathcal{B}\bar{u} = \int_{\mathbf{R}} e^z \bar{u}(t, x - z)v(z)dz.$$

To numerically solve system (2.18), we restrict our problem on a bounded computational domain $x \in \Omega \equiv [\underline{x}, \bar{x}]$ and $t \in [0, T]$. We further simplify the complementarity system (2.18) by defining $u = \bar{u} - \psi$. Then system (2.18) is equivalent to the following complementarity system. For $t \in [0, T]$ and $x \in \Omega$,

$$\begin{aligned} &\frac{\partial u}{\partial t} - \mathcal{A}u - \mathcal{B}u - \mathcal{A}\psi - \mathcal{B}\psi \geq 0, \\ &u \geq 0, \\ &\left(\frac{\partial u}{\partial t} - \mathcal{A}u - \mathcal{B}u - \mathcal{A}\psi - \mathcal{B}\psi\right) u = 0, \end{aligned} \tag{2.19}$$

with initial condition

$$u(x, 0) = \psi(x), \ x \in \Omega. \tag{2.20}$$

Throughout this thesis, we solve complementarity systems (2.10), (2.11) and (2.19) obtained from the forward option pricing problems by discretizing them as linear complementarity problems. Different discretization methods have been proposed by researchers to transform the complementarity systems into linear complementarity problems. We mainly use two popular discretization schemes, a finite difference method and a finite element method in this thesis. We demonstrate the discretization schemes by applying them to the complementarity systems obtained under the Dupire system and the local volatility model. Before presenting the details of the finite difference method and the finite element method, we begin with the review of a linear complementarity problem (LCP) and some properties of the LCP which are closely related to our pricing problems. Based on these properties, we can analyze the existence and the uniqueness of the solutions to the corresponding LCPs, and develop effective algorithms to obtain the approximate solutions.

2.2.2 Linear complementarity problem and related properties

In the mathematical optimization theory, a LCP searches for a vector $z \in \mathbf{R}^n$ with a given vector $q \in \mathbf{R}^n$ and a given matrix $M \in \mathbf{R}^{n\times n}$ such that

$$\begin{aligned} & z \geq 0, \\ & q + Mz \geq 0, \\ & z^T(q + Mz) = 0. \end{aligned} \tag{2.21}$$

For simplicity, we refer the above LCP as $LCP(q, M)$, and denote the solution set of the $LCP(q, M)$ as $SOL(q, M)$.

The structure of the associated matrix M plays an important role to the solution of the $LCP(q, M)$. There is a special class of matrices such that if M belongs to this special class of matrices, then for any $q \in \mathbf{R}^n$, there is a unique solution to the $LCP(q, M)$. This special class of matrices are called P-matrices. We now present the definition of a P-matrix and its key properties [CPS09, Section 3.3] as follows.

Definition 2.2.1 (P-matrix). A matrix $M \in \mathbf{R}^{n\times n}$ is said to be a P-matrix if the determinants of all principal sub-matrices of M are positive.

The key characterizations of a P-matrix are summarized in the following theorem:

Theorem 2.2.2. *Let $M \in \mathbf{R}^{n\times n}$, the following statements are equivalent:*

(1) A matrix M is a P-matrix.

(2) M reverses the sign of no non-zero vector, i.e.,

$$z_i(Mz)_i \leq 0 \ \textit{ for all}\, i = 1, 2, \ldots, n \Rightarrow z = 0. \tag{2.22}$$

(3) All real eigenvalues of M and its principal sub-matrices are positive.

(4) The $LCP(q, M)$ has a unique solution for all constant vectors $q \in \mathbf{R}^n$; moreover, the solution $z(q)$ as a function of q is Lipschitz continuous.

Here, by a principal sub-matrix of M, we mean any matrix of the form $M_{\alpha\alpha}$ with $\alpha \subseteq \{1, 2, \ldots, n\}$. Detailed proof of Theorem 2.2.2 can be found in [CPS09, Chapter 3]. Notice that from the last statement of the above theorem, we know that the P-property of a matrix M guarantees the existence and the uniqueness of the solution to the $LCP(q, M)$. In addition, we can decide whether a matrix is a P-matrix or not by using the third statement of Theorem (2.2.2). In this work, we choose discretization steps properly so that the associated matrices of all LCPs transformed from the complementarity systems are P-matrices. Particularly, they belong to a special subclass of P-matrices, strictly row diagonally dominant matrices.

Definition 2.2.3 (row diagonally dominant). A matrix $M \in \mathbf{R}^{n \times n}$ is row diagonally dominant if for $i = 1, 2, \ldots, n$,

$$\mid M_{ii} \mid \geq \sum_{j \neq i} \mid M_{ij} \mid .$$

When the strict inequality holds for all i, the matrix M is called a strictly row diagonally dominant matrix.

For any row strictly diagonally dominant matrix, we have the following theorem:

Theorem 2.2.4. *If $M \in \mathbf{R}^{n \times n}$ is a strictly row diagonally dominant matrix with positive diagonal entries, then M is a P-matrix; moreover for any vector $q \in \mathbf{R}^n$, the $LCP(q, M)$ has a unique solution.*

Based on Theorem 2.2.4, we know that if we choose the discretization steps properly, the associated matrix of the LCP obtained from the pricing problem can be a strictly row diagonally dominant matrix with positive diagonal entries. Hence, there is a unique solution to the corresponding LCP for any payoff function and boundary condition.

Now we have introduced the definition of LCP and its related properties, we present two popular discretization schemes, the finite difference method and the finite element method, used for the transformation of complementarity systems into linear complementarity problems in rest of this section. We illustrate each discretization scheme by applying them to our complementarity systems obtained in Section 2.2.1. Notice that in our study, we mainly use the finite difference method to discretize the complementary system derived under the local volatility model (2.11), and the finite element method to discretize the complementarity system derived under the Dupire system (2.19). We start our illustration with the finite difference method.

2.2.3 Linear complementarity system of the local volatility model

We follow [HP00] and resort to a finite difference method for approximating the partial differential complementarity system (2.11). We first truncate the state variable S into a finite range $[0, S_{\max}]$, and construct a regular grid of size $R \times Q$ on the closed region $[0, S_{\max}] \times [0, T]$. Here, $Q > 0$ is the discretization number for the space variable S, and $R > 0$ is the discretization number for the time variable t. Let

$$\delta t = \frac{T}{Q}, \text{ and } \delta S = \frac{S_{\max}}{R},$$

then any grid point on the above grid can be defined as (S_i, t_m) with

$$S_i = i\delta S, \ \ i = 0, 1, \ldots, Q,$$
$$t_m = m\delta t, \ \ m = 0, 1, \ldots, R.$$

Similar to [HP00], we use the following expressions to approximate the first and second partial derivatives appeared in system (2.11):

$$\begin{aligned}
\frac{\partial V(S,t)}{\partial t} =& \frac{V(S,t+\delta t) - V(S,t)}{\delta t} + O(\delta t), \\
\frac{\partial V(S,t)}{\partial S} =& \theta \frac{V(S+\delta S,t) - V(S-\delta S,t)}{2\delta S} \\
&+ (1-\theta)\frac{V(S+\delta S,t+\delta t) - V(S-\delta S,t+\delta t)}{2\delta S} + O\left((\delta S)^2\right), \\
\frac{\partial^2 V(S,t)}{\delta S^2} =& \theta \frac{V(S+\delta S,t) - 2V(S,t) + V(S-\delta S,t)}{\delta S^2} \\
&+ (1-\theta)\frac{V(S+\delta S,t+\delta t) - 2V(S,t+\delta t) + V(S-\delta S,t+\delta t)}{\delta S^2} + O\left((\delta S)^2\right),
\end{aligned} \tag{2.23}$$

with $\theta \in [0,1]$ being a given parameter. The finite difference approximation is called an explicit approximation when $\theta = 0$. In this case, the discretization scheme actually uses three points, $V(S, t+\delta t)$, $V(S-\delta S, t+\delta t)$ and $V(S+\delta S, t+\delta t)$ of time $t+\delta t$ to approximate $V(S,t)$ of time t. Such a method can be viewed as a special case of pricing American options on a trinomial tree. The approximation is called an implicit approximation when $\theta = 1$. Notice that when $\theta = 1$, three points, $V(S,t)$, $V(S-\delta S,t)$, $V(S+\delta S,t)$ of time t are used to approximate the first and second partial derivatives of $V(S,t)$ with respect to variable S in system (2.11). We call the approximation a Crank-Nicolson approximation when $\theta = 0.5$. This method can be viewed as a mix of the explicit and the implicit approximations. Six points on the grid are involved when using the Crank-Nicolson approximation. In Figure 2.1, we show the difference between the implicit and the explicit approximations.

To fully solve the linear complementarity problem obtained from system (2.11), we need to impose some boundary conditions on the grid points $V(0, t_m)$ and $V(S_{\max}, t_m)$ for $m = 0, 1, \ldots, R-1$. In our study, we mainly consider pricing American put options and impose the following boundary conditions on the American put options. For $m = 0, 1, \ldots, R-1$,

$$V(0, t_m) = \Psi(0) = K,$$
$$V(S_{\max}, t_m) = \Psi(S_{\max}) = (K - S_{\max})^+.$$

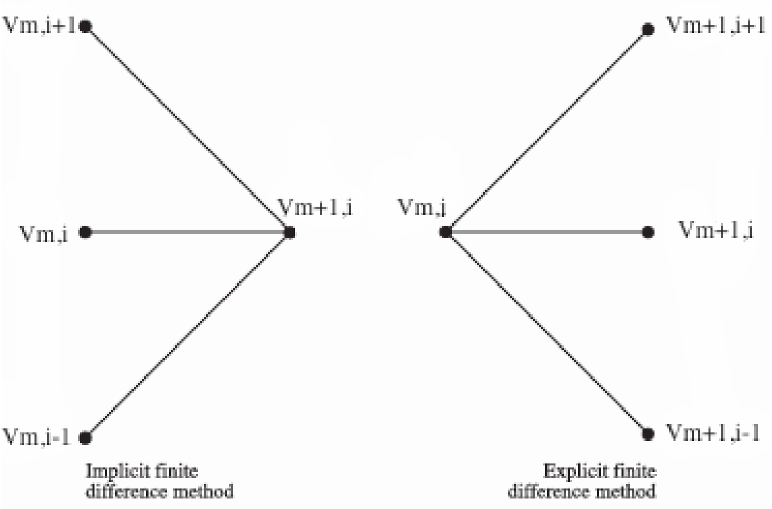
Vm,i+1
Vm,i
Vm,i-1
Vm+1,i
Implicit finite difference method
Vm+1,i+1
Vm,j
Vm+1,i
Vm+1,i-1
Explicit finite difference method

Figure 2.1: Difference between the implicit and explicit finite difference methods.

Combing the approximation (2.23), the terminal and boundary conditions on $V(S,t)$, we now can solve for the unknown option values $V(i\delta S, m\delta t)$, $i = 1, 2, \dots, Q-1$ and $m = 0, 1, \dots, R-1$. Denote

$$\begin{aligned} V_i^m &\equiv V(i\delta S, m\delta t), \\ \sigma_i^m &\equiv \sigma(i\delta S, m\delta t), \end{aligned}$$

for $i = 0, 1, \dots, Q$, and $m = 0, 1, \dots, R$. Together with the boundary and the terminal conditions

$$\begin{aligned} V_0^m &= \Psi(0) = K, \\ V_Q^m &= \Psi(S_{\max}) = (K - S_{\max})^+, \\ V_i^R &= \Psi(i\delta S) = \max(K - i\delta S, 0),\ i = 1, 2, \dots, Q, \end{aligned}$$

we approximate system (2.11) by R LCPs, each of order $(Q-1)$. Specifically, at time $t = m\delta t$, $m = 0, 1, \dots, R-1$, we have

$$0 \le \mathbf{V}^m - \mathbf{\Psi}^m \perp (\mathbf{d}^m + \mathbf{q}^{m+1} + \mathbf{M}^m \mathbf{V}^m) \ge 0, \tag{2.24}$$

where

$$\mathbf{V}^m \equiv \begin{pmatrix} V_1^m \\ \vdots \\ V_{Q-1}^m \end{pmatrix}, \ \mathbf{d}^m \equiv \begin{pmatrix} d_1^m \\ \vdots \\ d_{Q-1}^m \end{pmatrix},$$

with entries

$$d_i^m = \begin{cases} \frac{1}{2}\left(\theta V_0^m + (1-\theta)V_0^{m+1}\right)\left(r - q - (\sigma_1^m)^2\right), & i = 1, \\ -\frac{1}{2}\left(\theta V_Q^m + (1-\theta)V_Q^{m+1}\right)\left((Q-1)(r-q) + (Q-1)^2(\sigma_{Q-1}^m)^2\right), & i = Q-1, \\ 0, & otherwise, \end{cases}$$

and $\mathbf{q}^{m+1} = \widehat{\mathbf{M}}^m \mathbf{V}^{m+1}$.

The associated matrices $\mathbf{M}^m$ and $\widehat{\mathbf{M}}^m$ can be calculated explicitly and take the following forms:

$$\mathbf{M}^m \equiv \begin{pmatrix} b_1 & c_1 & 0 & 0 & \dots & 0 \\ a_2 & b_2 & c_2 & 0 & \dots & 0 \\ 0 & a_3 & b_3 & c_3 & \dots & 0 \\ 0 & 0 & a_4 & b_4 & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & c_{Q-2} \\ 0 & 0 & 0 & \dots & a_{Q-1} & b_{Q-1} \end{pmatrix}, \quad \widehat{\mathbf{M}}^m \equiv \begin{pmatrix} \hat{b}_1 & \hat{c}_1 & 0 & 0 & \dots & 0 \\ \hat{a}_2 & \hat{b}_2 & \hat{c}_2 & 0 & \dots & 0 \\ 0 & \hat{a}_3 & \hat{b}_3 & \hat{c}_3 & \dots & 0 \\ 0 & 0 & \hat{a}_4 & \hat{b}_4 & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \hat{c}_{Q-2} \\ 0 & 0 & 0 & \dots & \hat{a}_{Q-1} & \hat{b}_{Q-1} \end{pmatrix},$$

with entries

$$\begin{aligned} a_i &\equiv -\frac{1}{2}\theta(\sigma_i^m)^2 i^2 + \theta\frac{(r-q)i}{2}, \\ b_i &\equiv \frac{1}{\delta t} + \theta(\sigma_i^m)^2 i^2 + r, \\ c_i &\equiv -\frac{1}{2}\theta(\sigma_i^m)^2 i^2 - \theta\frac{(r-q)i}{2}, \\ \hat{a}_i &\equiv -\frac{1}{2}(1-\theta)(\sigma_i^m)^2 i^2 + (1-\theta)\frac{(r-q)i}{2}, \\ \hat{b}_i &\equiv -\frac{1}{\delta t} + (1-\theta)(\sigma_i^m)^2 i^2, \\ \hat{c}_i &\equiv -\frac{1}{2}(1-\theta)(\sigma_i^m)^2 i^2 - (1-\theta)\frac{(r-q)i}{2}. \end{aligned}$$

Notice that both $\mathbf{M}^m$ and $\widehat{\mathbf{M}}^m$ are asymmetric tridiagonal matrices with entries dependent upon the finite difference method, the time variable t and the space variable S. This is because in the local volatility model, the volatility is no longer a constant but a function of the time and the space variables. Hence, the time and space variables affect the LCPs via the volatility function.

According to Theorem 2.2.4, the existence and the uniqueness of the solution to LCP (2.24) are determined by the structure of the associated matrix $\mathbf{M}^m$, $m = 0, 1, \ldots, R-1$. When

$$\frac{1}{\delta t} + r \geq Q|r - q|, \tag{2.25}$$

it is easy to verify that the matrix $\mathbf{M}^m$ is strictly row diagonally dominant, and hence a P-matrix for all $m = 0, 1, \ldots, R-1$ and $\theta \in [0, 1]$. Therefore, all the LCPs (2.24) are solvable and have unique solutions. Accordingly, the solution to the original option pricing problem exists and is unique. Throughout the thesis, we assume condition (2.25) is satisfied.

2.2.4 Linear complementarity system of the Dupire system

Another popular approach for transforming a complementarity system into a linear complementarity problem is through a finite element method. This method uses many simple element equations over small subdomains, named finite elements, to approximate a more complex equation over a larger domain. In this thesis, we follow a similar approach adopted by [FL08] and [FLMN10], and apply a finite element method to transform the complementarity system (2.19) into a linear complementarity problem.

To begin with, we restrict our problem on a bounded domain $\Omega = [\underline{x}, \bar{x}]$ and impose the following boundary condition on $\partial\Omega = \{\underline{x}, \bar{x}\}$:

$$u(t, x) = 0,\ t \in [0, T],\ x \in \partial\Omega.$$

We multiply both sides of the first inequality of system (2.19) by $w(x) - u(x)$. Here, $w(x) \geq 0$ is a test function. It is square-integrable on Ω, vanishes on $\partial\Omega$, and has a first derivative that is also square-integrable on Ω. We then integrate the above result over Ω and obtain the following system:

$$\left(\frac{\partial u}{\partial t}, w - u\right) + a(u, w - u) - b(u, w) + a(\psi, w - u) - b(\psi, w - u) \geq 0. \tag{2.26}$$

Here, $(\cdot, \cdot)$, the inner product in $\mathrm{L}^2(\Omega)$, is calculated as

$$\left(\frac{\partial u}{\partial t}, w - u\right) = \int_{\underline{x}}^{\bar{x}} \frac{\partial u}{\partial t}(w - u)dx,$$

and the bilinear forms $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ are calculated as

$$
\begin{aligned}
a(u,w) &= \frac{\sigma^2}{2}\int_{\underline{x}}^{\bar{x}} \frac{\partial u}{\partial x}\frac{\partial w}{\partial x}dx + (\mu + \frac{\sigma^2}{2})\int_{\underline{x}}^{\bar{x}} \frac{\partial u}{\partial x} w dx + (r-\mu+\lambda)\int_{\underline{x}}^{\bar{x}} uwdx,\\
b(u,w) &= \int_{\underline{x}}^{\bar{x}}\int_{\mathbf{R}} e^z u(x-z)w(x)v(z)dxdz.
\end{aligned}
$$

Notice that we use the fact $(\mathcal{A}u, w-u) = -a(u, w-u)$ in the above integration.

We then apply a set of basis functions to the complementarity system (2.26) and transform it into a variational inequality defined by

Definition 2.2.5 (variational inequality). For a given subset $\Gamma \subseteq \mathbf{R}^n$ and a mapping $G : \Gamma \to \mathbf{R}^n$. The variational inequality is to find a vector $z \in \Gamma$ such that

$$(y-z)^T G(z) \geq 0, \ \forall y \in \Gamma.$$

A more detailed description of the variational inequality and its related properties can be found in [FP03b, Chapter 1]. To construct the set of the basis functions, we divide the interval Ω into $Q+1$ subintervals of equal length $h = (\underline{x} - \bar{x})/(Q+1)$. Denote $x_i = \underline{x} + ih$, $i = 0, 1, \ldots, Q$ as the nodes in Ω. We define the piecewise-linear basis functions by

$$
\phi_{h,i}(x) \equiv \begin{cases} (x - x_{i-1})/h, & x_{i-1} \leq x \leq x_i,\\ (x_{i+1} - x)/h, & x_i < x \leq x_{i+1},\\ 0, & x \notin [x_{i-1}, x_{i+1}], \end{cases}
$$

for $i = 1, 2, \ldots, Q$. According to the definition of the basis functions $\phi_{h,i}$, we know that each basis function takes value 1 at node x_i and zero at all other nodes. Given the above definition of the basis functions, it is easy to verify that all the basis functions and their linear combinations are test functions. In Figure 2.2, we plot three sample basis functions at nodes $x_i = 1, 2, 3$ with $h = 1$ to provide readers a clear view of the basis functions $\phi_{h,i}$.

In the finite element method, we search for a finite element approximation u_h to the solution of system (2.19). Hence, we express u_h as a linear combination of the basis functions with time-dependent coefficients.

$$u_h(t,x) = \sum_{i=1}^{Q} u_i(t)\phi_{h,i}(x), \ \ t \in [0,T].$$

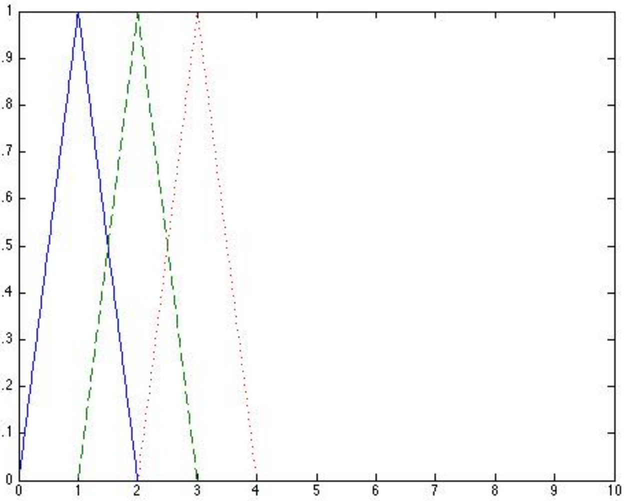

Figure 2.2: $\phi_{1,i}(x)$ for $x_i = 1, 2, 3$.

Notice that under the construction of u_h, the boundary condition (2.20) is satisfied naturally. Denote

$$u(t) = (u_1(t), \ldots, u_Q(t))^T,$$
$$\phi_h(x) = (\phi_{h,1}(x), \ldots, \phi_{h,Q}(x))^T,$$

then

$$u(t,x) = u(t)^T \phi_h(x). \tag{2.27}$$

The Q-dimensional vector $u(t)$ formed by the time-dependent coefficients are our approximate time values of the options with strike prices $\{Se^{x_i}\}$, $i = 1, 2, \ldots, Q$ at time t. Here, by the time value of an option, we mean the difference between an option price and its intrinsic value. The second inequality of system (2.19) further guarantees that $u(t) \geq 0$. Let the test function $w(x) \geq 0$ be a linear combination of the above constructed basis functions with coefficient vector $w = (w_1, \ldots, w_Q)^T$. Then

$$w(x) = w^T \phi_h(x). \tag{2.28}$$

Substitute (2.27) and (2.28) into (2.26), we have

$$
\begin{aligned}
&\left(\frac{\partial u}{\partial t}, w-u\right) + a(u, w-u) - b(u, w) + a(\psi, w-u) - b(\psi, w-u) \\
&= (u'(t)^T\phi_h, w^T\phi_h - u(t)^T\phi_h) + a\left(u(t)^T\phi_h, w^T\phi_h - u(t)^T\phi_h\right) - b\left(u(t)^T\phi_h, w^T\phi_h - u(t)^T\phi_h\right) \\
&+ a\left(\phi_h, w^T\phi_h - u(t)^T\phi_h\right) - b\left(\phi_h, w^T\phi_h - u(t)^T\phi_h\right) \\
&\geq 0.
\end{aligned}
$$

Since for any $w \geq 0$, the above condition holds. Therefore, $u(t)$ is the solution to the variational inequality

$$
(w - u(t))^T \cdot (\mathbf{M}\cdot u'(t) + \mathbf{A}\cdot u(t) - \mathbf{B}\cdot u(t) + \mathbf{F}) \geq 0,\ w \geq 0. \tag{2.29}
$$

Here, $(u'(t))^T = (\frac{du_1}{dt}, \ldots, \frac{du_Q}{dt})$; $\mathbf{M} = (\mathbf{M}_{ij})$ is the mass matrix with $\mathbf{M}_{ij} = (\phi_{h,j}, \phi_{h,i})$; $\mathbf{A} = (\mathbf{A}_{ij})$ is the stiffness matrix with $\mathbf{A}_{ij} = a(\phi_{h,j}, \phi_{h,i})$; $\mathbf{B} = (\mathbf{B}_{ij})$ is the jump matrix with $\mathbf{B}_{ij} = b(\phi_{h,j}, \phi_{h,i})$; and $\mathbf{F} = (\mathbf{F}_1, \ldots, \mathbf{F}_Q)$ is the load vector with $\mathbf{F}_i = a(\psi, \phi_{h,i}) - b(\psi, \phi_{h,i})$. The last term $\mathbf{F}$ can be approximated by the finite element interpolants, the matrices $\mathbf{M}$ and $\mathbf{A}$ can be computed analytically.

The variational inequality (2.29) is equivalent to the following complementarity system:

$$
0 \leq u(t) \perp \mathbf{M}\cdot u'(t) + \mathbf{A}\cdot u(t) - \mathbf{B}\cdot u(t) + \mathbf{F} \geq 0,
$$

which can be further solved numerically via a discretization of the time variable t. Divide the time interval $[0, T]$ into R equal subintervals with length $dt = T/R$. Let $t_j = jdt,\ j = 0, 1, \ldots, R$ be the temporal nodes and denote $u(t_j)$ by u^j. By applying the explicit finite difference method to system (2.29), we obtain a sequence of LCPs of the following form:

$$
\begin{aligned}
&(u^j)^T \cdot \left((\mathbf{M} + dt\mathbf{A})u^j - (\mathbf{M} + dt\mathbf{B})u^{j-1} + dt\mathbf{F}\right) = 0, \\
&(\mathbf{M} + dt\mathbf{A})u^j - (\mathbf{M} + dt\mathbf{B})u^{j-1} + dt\mathbf{F} \geq 0, \\
&u^j \geq 0,\ 1 \leq j \leq R,
\end{aligned} \tag{2.30}
$$

with an initial condition

$$
u^0 = 0.
$$

Here, we discretize the time variable t with the explicit finite difference method to simplify the calculation of the associated matrix $\mathbf{M} + dt\mathbf{A}$ in LPCs (2.30). The matrices $\mathbf{M}$ and $\mathbf{A}$ can be calculated explicitly as

$$\mathbf{M} \equiv \frac{h}{6}\begin{pmatrix} 4 & 1 & \dots & 0 \\ 1 & 4 & \dots & \vdots \\ \vdots & \vdots & \ddots & 1 \\ 0 & \dots & 1 & 4 \end{pmatrix}, \quad \mathbf{A} \equiv \begin{pmatrix} a_0 & a_1 & \dots & 0 \\ a_{-1} & a_0 & \dots & \vdots \\ \vdots & \vdots & \ddots & a_1 \\ 0 & \dots & a_{-1} & a_0 \end{pmatrix},$$

with

$$\begin{aligned} a_0 &= \frac{2}{3}(r - \mu + \lambda)h + \frac{\sigma^2}{h}, \\ a_1 &= \frac{1}{2}(\mu + \frac{\sigma^2}{2}) + \frac{1}{6}(r - \mu + \lambda)h - \frac{\sigma^2}{2h}, \\ a_{-1} &= -\frac{1}{2}(\mu + \frac{\sigma^2}{2}) + \frac{1}{6}(r - \mu + \lambda)h - \frac{\sigma^2}{2h}. \end{aligned}$$

Moreover, the jump matrix $\mathbf{B}$ is a Toeplitz matrix, in which all the descending diagonals from left to right are constant. For simplicity, we denote $\phi_j \equiv \phi_{h,j}$. Since

$$\begin{aligned} B_{ij} =& b(\phi_j, \phi_i) \\ =& \int_{\underline{x}}^{\bar{x}} \int_{\mathbf{R}} e^z \phi_j(x - z)\phi_i(x)v(z)dxdz. \\ =& -\int_{x_{i-1}}^{x_{i+1}} \int_{x_{j-1}}^{x_{j+1}} e^{x-y}\phi_j(y)\phi_i(x)v(x - y)dydx \\ =& -\int_{-1}^{1} \int_{-1}^{1} e^{(u-w+i-j)h} h^2 \phi_{1,0}(u)\phi_{1,0}(w)v((u - w + i - j)h)dudw \end{aligned}$$

Therefore, B_{ij} only depends on $i - j$ and the jump matrix $\mathbf{B}$ is a Toeplitz matrix. $2Q - 1$ entries need to be calculated to fully characterize the matrix $\mathbf{B}$. In this work, we follow the same approach employed in [FL08] to approximate the entry B_{ij} using the finite element interpolants $(\mathcal{B}\phi_j, \phi_i)$.

$$(\mathcal{B}\phi_j, \phi_i) = \frac{1}{6}h\mathcal{B}\phi_j(x_{i-1}) + \frac{2}{3}h\mathcal{B}\phi_j(x_i) + \frac{1}{6}h\mathcal{B}\phi_j(x_{i+1}),$$

with

$$\mathcal{B}\phi_j(x_i) = \int_{\mathbf{R}} e^z \phi_j(x_i - z)v(z)dz,$$

for $i, j = 1, 2, \dots, Q$. It is also easy to verify that for any $i, j = 1, 2, \dots, Q$, $\mathcal{B}\phi_j(x_i)$ depends only on $i - j$.

$$\begin{aligned}
\mathcal{B}\phi_j(x_i) &= \int_{\mathbf{R}} e^z \phi_j(x_i - z) v(z) dz \\
&= \int_{\mathbf{R}} e^z \phi_{1,0}\left(\frac{x_i - x_j - z}{h}\right) v(z) dz \\
&= -h \int_{-1}^{1} e^{(i-j-y)h} \phi_{1,0}(y) v\left((i-j-y)h\right) dy.
\end{aligned}$$

Hence, the approximate matrix of $\mathbf{B}$ is a Toeplitz matrix as well.

For Kou's jump diffusion model, $\mathcal{B}\phi_j(x_i)$ can be calculated explicitly by

Case 1: when $i = j$

$$\begin{aligned}
\mathcal{B}\phi_j(x_i) &= \lambda \int_{\mathbf{R}} e^z \phi_j(x_j - z) v(z) dz \\
&= \lambda \int_{\mathbf{R}} e^z \phi_{1,0}(-z/h) v(z) dz \\
&= \lambda \left(\int_{-h}^{0} \eta_2 (1-p) e^{(1+\eta_2)z} (1 + z/h) dz + \int_{0}^{h} \eta_1 p e^{(1-\eta_1)z} (1 - z/h) dz \right) \\
&= \lambda p \eta_1 \left(-\frac{1}{1-\eta_1} + \frac{e^{(1-\eta_1)h} - 1}{(1-\eta_1)^2 h} \right) + \lambda (1-p) \eta_2 \left(\frac{1}{1+\eta_2} - \frac{1 - e^{-(1+\eta_2)h}}{(1+\eta_2)^2 h} \right).
\end{aligned}$$

Case 2: when $i \geq j + 1$

$$\begin{aligned}
\mathcal{B}\phi_j(x_i) &= \lambda \int_{\mathbf{R}} e^z \phi_j(x_i - z) v(z) dz \\
&= \lambda \int_{\mathbf{R}} e^z \phi_{1,0}(i - j - z/h) v(z) dz \\
&= \lambda p \eta_1 \left(\int_{(i-j-1)h}^{(i-j)h} e^{(1-\eta_1)z} (1 - i + j + z/h) dz + \int_{(i-j)h}^{(i-j+1)h} e^{(1-\eta_1)z} (1 + i - j - z/h) dz \right) \\
&= \lambda p \eta_1 \left(e^{(1-\eta_1)(i-j+1)h} + e^{(1-\eta_1)(i-j-1)h} - 2 e^{(1-\eta_1)(i-j)h} \right) / \left((1-\eta_1)^2 h \right).
\end{aligned}$$

Case 3: when $i \leq j - 1$

$$\begin{aligned}
\mathcal{B}\phi_j(x_i) &= \lambda \int_{\mathbf{R}} e^z \phi_j(x_i - z) v(z) dz \\
&= \lambda \int_{\mathbf{R}} e^z \phi_{1,0}(i - j - z/h) v(z) dz \\
&= \lambda (1-p) \eta_2 \left(\int_{(i-j-1)h}^{(i-j)h} e^{(1+\eta_2)z} (1 - i + j + z/h) dz + \int_{(i-j)h}^{(i-j+1)h} e^{(1+\eta_2)z} (1 + i - j - z/h) dz \right) \\
&= \lambda (1-p) \eta_2 \left(e^{(1+\eta_2)(i-j+1)h} + e^{(1+\eta_2)(i-j-1)h} - 2 e^{(1+\eta_2)(i-j)h} \right) / \left((1+\eta_2)^2 h \right).
\end{aligned}$$

With the above transformation, we now can solve the original pricing problems by solving a collection of LCPs. We will introduce four existing popular LCP solvers in the next chapter. In addition, we will use them to solve the LCPs derived in this chapter.

Chapter 3

Solvers for linear complementarity problems

As suggested in Chapter 2, a fast LCP solver which solves a sequence of LCPs efficiently, is of great importance to option pricing problems. A variety of algorithms have been proposed by researchers to solve LCPs. For instance, a recent study by Borici and Luethi [BL05] reported promising numerical results of solving LCPs with Z-matrices by using a variant of the simplex-like method. Ikonen and Toivanen [IT04] developed four different methods—a projected multi-grid method, an operator splitting method, a penalty method, and a component-wise splitting method—for solving LCPs. In general, the LCP solvers can be categorized into two groups, pivoting methods and iterative methods. Pivoting methods typically end in finite steps and are similar to the simplex methods for solving linear programs; iterative methods do not terminate in finite steps, but they usually yield a sequence of approximations to the solutions of LCPs.

In this thesis, we mainly discuss four existing popular LCP solvers, including a projected successive over-relaxation (PSOR) method, a two phase active-set method, a parametric pivoting method and a semi-smooth Newton method. A rich literature of the PSOR method and a parametric pivoting method can be found in [CPS09, Chapter 4] of Cottle, Pang and Stone. A detailed description of the semi-smooth Newton method and its related theoretical results can be found in [FP03a, Chapter 9] by Facchinei and Pang. The two phase active-set method is first proposed by Morales, Nocedal and Smelyanskiy in [MNS08], and used later by Feng, Linetsky, Morales and Norcedal for solving LCPs related to the pricing problems of American options in [FLMN10]. The convergence of a variant of the algorithm is provided in a subsequent paper by Robinson, Feng, Norcedal and Pang in [RFNP11].

Before we give a complete description of the four algorithms discussed above, we first briefly discuss the different utilization of pivoting methods and iterative methods. In general, we use each type of methods to solve LCPs with different matrix structures. As derived in Section 2.2.3 and Section 2.2.4, both the finite difference method and the finite element method transform a complementarity system into a sequence of linear complementarity problems with tridiagonal matrices. We can take advantage of the sparsity of the associated matrices by employing an iterative method. This is because an iterative method maintains the tridiagonal structures of the matrices in the LCPs throughout the computation. On the other hand, a pivoting method is in general not efficient for solving LCPs with sparse matrices. Usually pivoting methods destroy the sparsity of the matrices in a few steps. Therefore, with more non-zero entries, the computational cost of each step will increase for the later steps of pivoting methods. However, for a LCP with a tridiagonal matrix, we can still take advantage of the sparsity by employing the Gaussian elimination method during the pivot step.

The remainder of this chapter is organized as follows. In Section 3.1–Section 3.4, we discuss the four existing algorithms: the PSOR method, the two phase active-set method, the semi-smooth Newton method and the parametric pivoting method in details. In Section 3.5, we test the numerical performance of the above four methods for solving LCPs derived under Kou's jump diffusion model and the Dupire system. As discussed in the previous paragraph, pivoting methods might not work well for solving LCPs with sparse matrices. Therefore, we only test the parametric pivoting method with Kou's jump diffusion model. The three iterative methods, the PSOR method, the two phase active-set method and the semi-smooth Newton method will be tested with both Kou's jump diffusion model and the Dupire system.

3.1 Projected successive over-relaxation method

One of the most widely used LCP solvers for sparse problems is a projected successive over-relaxation (PSOR) method. The PSOR method is a special case of matrix splitting methods for solving systems of linear equations. Similar to a general matrix splitting method, we split the associated matrix M of the $LCP(q, M)$ into the sum of a lower triangular matrix $B = L + \omega^{-1}D$ and a upper triangular matrix $C = U + (1 - \omega^{-1})D$. Here, matrices L, D and U are the strictly lower triangular part, the diagonal part and the strictly upper triangular part of the matrix M, and ω is chosen from $(0, 2)$.

Given the matrix splitting, we then search for the solution of an approximate $LCP(q^\nu, B)$ at each iteration ν with

$$q^\nu = q + Cz^\nu.$$

Due to the special construction of the matrix B in the PSOR method, if the diagonal entries of the matrix M are all positive, then all the diagonal entries of the matrix B are positive as well. Under this condition, the sub-problem $LCP(q^\nu, B)$ at each iteration is well-defined and can be efficiently solved. In the case of the LCPs derived under the jump diffusion models, the matrices are all strictly row diagonally dominant matrices with positive diagonal entries. Hence, the PSOR method is well-defined in our problems. Once we find the solution of the sub-problem $LCP(q^\nu, B)$, we record it as $z^{\nu+1}$ and use it to generate the sub-problem $LCP(q^{\nu+1}, B)$ for the next iteration $\nu + 1$.

Like other iterative methods, the PSOR method does not terminate in finite steps. Therefore, we terminate the PSOR method when a certain condition is satisfied by z^ν. Such a termination criterion is defined in terms of the residual function of the $LCP(q, M)$.

$$r(z, q, M) = || \min(Mz + q, z)||_\infty.$$

Here, $\min(a, b)$ is a vector with the i-th entry being $\min(a_i, b_i)$, and $\|a\|_\infty$ represents the infinity norm of the corresponding vector a. It is easy to verify that the residual function $r(z, q, M) = 0$ if and only if z solves the $LCP(q, M)$ exactly. Hence, we use the residual function to form the stopping criterion in our implementation. More specifically, we stop the PSOR method when

$$r(z^\nu, q, M) \leq \epsilon_{\text{sor}},$$

where $\epsilon_{\text{sor}} > 0$ is the given tolerance level of the algorithm. At the termination of the PSOR method, we use z^ν as our approximate solution to the original $LCP(q, M)$. Details of the PSOR method are presented in Algorithm 1.

Algorithm 1: PSOR method

```
Input: matrix $M$ of size $n \times n$, vector $q$ of size $n \times 1$, a positive scalar $\omega \in (0, 2)$;
initialize $z^0 \geq 0$, set $k \leftarrow 0$;
repeat
    for $i = 1, \dots, n$ do
        $z_i^{0.5} = -\frac{1}{M_{ii}}(q_i + \sum_{j<i} M_{ij} z_j^{k+1} + \sum_{j>i} M_{ij} z_j^k)$;
        $\gamma_i = z_i^k + \omega(z_i^{0.5} - z_i^k)$;
        $z_i^{k+1} = \max\{0, \gamma_i\}$.
    $k \leftarrow k + 1$;
until a stopping test is satisfied;
```

The advantages of the PSOR method are that it is simple to implement and has a small computational cost per iteration due to the splitting of the matrix M. Especially when encountering LCPs with matrices of sparse structures, this method performs quite efficiently. The main drawback of the PSOR method is its slow convergence. Thus, a two phase active-set method is proposed to overcome the slow convergence of the PSOR method by finding a good approximation of the solution in a reduced space.

3.2 Two phase active-set method

The two phase active-set method, first introduced in [MNS08], is an iterative method which combines iterations of the PSOR method with reduced-space steps. In its first phase, we run several PSOR iterations to obtain an initial guess of the active set of $SOL(q, M)$. Here, by the active set, we mean the index set where $z_i > 0$ for $z \in SOL(q, M)$. We then perform a reduced-space phase where the components that do not belong to the active set of z are kept at zero, and all the active components are updated by the solution of a system of linear equations. We repeat the loop of the PSOR iterations and the reduced-space iterations until we find a satisfactory result.

Now we describe the two phase active-set method in more details. At each loop k, we run k_{sor} iterations of the PSOR method, and record the corresponding result as z^k. Define $\alpha = \{i \mid z_i^k > 0 \text{ for } i = 1, \dots, n\}$ as the associated active set of z^k, and $\bar{\alpha} = \{i \mid z_i^k = 0 \text{ for } i = 1, \dots, n\}$ as the associated inactive set. We search for the next approximation to the solution of the $LCP(q, M)$ by fixing the zero components of z^k, and updating the non-zero components of z^k with the solution $\hat{z}_\alpha^k$ of the system of linear equations

$$M_{\alpha\alpha}\hat{z}_\alpha^k + q_\alpha = 0. \tag{3.1}$$

The advantage of using this approximation is that when the active set of z^k is the same as the active set of the solution z to the $LCP(q, M)$, z^{k+1} will be equal to z after the update. This achieves the goal of accelerating the convergence of the PSOR method.

It is worth mentioning that we make a few modifications to $\hat{z}_\alpha^k$, which is the solution to system (3.1), before we use it to update z^{k+1}. Notice that some components of $\hat{z}_\alpha^k$ might be negative, which conflict the non-negative condition of system (2.21). Therefore, we apply a projection on $\hat{z}_\alpha^k$ to the non-negative orthant before we use it to update z^{k+1}.

$$\hat{z}_\alpha^k \leftarrow \max(0, \hat{z}_\alpha^k).$$

Notice that because of the projection, some elements of $\hat{z}_\alpha^k$ might become zero. In this case, we can further apply the reduced-space method to a problem of form (3.1). We repeat the reduce-space method until the number of the negative components of $\hat{z}_\alpha^k$ is smaller than a prescribed value. When this condition is satisfied, we assume that the prediction of the active set changes little and can be used as an approximation of the true active set to update z^{k+1}. The algorithm is described in Algorithm 2.

The success of the two phase active-set method relies heavily on the repeated application of the reduced-space phase. It accelerates the estimation of the optimal active set, and therefore potentially leads to a fast

```
Algorithm 2: Two phase active-set method
  Input: matrix $M$ of size $n \times n$, vector $q$ of size $n \times 1$, integers $\Delta_{\text{ac}} > 0$ and $k_{\text{sor}} > 0$, and a positive
         scalar $\omega \in (0, 2)$;
  initialize $z^0 \geq 0$, set $k \leftarrow 0$;
  repeat
      perform $k_{\text{sor}}$ iterations of the PSOR method (Algorithm 1) starting with $z^k$;
      update $z^k$ with the result obtained from the PSOR iterations;
      repeat
          let $\alpha = \{i : z_i^k > 0\}$ denote the set of active variables at $z^k$;
          solve the following reduced linear system for $\hat{z}_\alpha^k$:
              $$M_{\alpha\alpha}\hat{z}_\alpha^k + q_\alpha = 0.$$
          set $\hat{z}_\alpha^k \leftarrow \max(0, \hat{z}_\alpha^k)$ and $nz \leftarrow$ number of zero components in $\hat{z}_\alpha^k$;
          if $nz \geq \Delta_{\text{ac}}$ then
              replace the components of $z^k$ that correspond to the set $\alpha$ with the elements of $\hat{z}_\alpha^k$.
          else
              set $z^{k+1} = 0$ and then set $z_\alpha^{k+1} \leftarrow \hat{z}_\alpha^{k+1}$;
              break;
      until;
      $k \leftarrow k + 1$;
  until a stopping test is satisfied;
```

convergence rate.

3.3 Semi-smooth Newton method

Semi-smooth Newton method is an iterative method, designed especially for solving a general complementarity problem $CP(F)$ of the following form:

$$0 \leq z \perp F(z) \geq 0. \tag{3.2}$$

Here, $F(z)$ is a continuous and differentiable function on $\mathbf{R}^n$. The semi-smooth Newton method is mainly developed based on an equivalent reformulation of the original complementarity system (3.2). In this section, we first present the equivalent reformulation of the $CP(F)$, and then develop the semi-smooth Newton method based on the reformulation.

Given a $CP(F)$, we define the following function $\mathbf{F}_{\text{FB}}$ on $z \in \mathbf{R}^n$:

$$\mathbf{F}_{\text{FB}}(z) \equiv \begin{pmatrix} \phi_{\text{FB}}(z_1, F_1(z)) \\ \vdots \\ \phi_{\text{FB}}(z_n, F_n(z)) \end{pmatrix},$$

with

$$\phi_{\mathrm{FB}}(a,b) \equiv \sqrt{a^2+b^2} - (a+b),\ a,b \in \mathbf{R}^+.$$

The function $\mathbf{F}_{\mathrm{FB}}$ is called a Fischer-Burmeister (FB) C-function. It was first proposed by Fischer in [Fis92], and used later by De Luca, Facchinei, and Kanzow in [DLFK96] for the reformulation of the nonlinear complementarity problems. In case of a $LCP(q,M)$ with a matrix M of size n, the corresponding $\mathbf{F}_{\mathrm{FB}}$ can be explicitly written as

$$\mathbf{F}_{\mathrm{FB}}(z) \equiv \begin{pmatrix} \phi_{\mathrm{FB}}\left(z_1, (q+Mz)_1\right) \\ \vdots \\ \phi_{\mathrm{FB}}\left(z_n, (q+Mz)_n\right) \end{pmatrix}.$$

Notice that the function $\mathbf{F}_{\mathrm{FB}}$ has the property that

$$\mathbf{F}_{\mathrm{FB}}(z) = 0 \Leftrightarrow z \in SOL(q,M).$$

Therefore, the equation $\mathbf{F}_{\mathrm{FB}}(z) = 0$ provides an equivalent formulation of the $LCP(q,M)$. By solving $\mathbf{F}_{\mathrm{FB}}(z) = 0$, we find a solution to the $LCP(q,M)$. Based on this property of $\mathbf{F}_{\mathrm{FB}}$, we can design a semi-smooth Newton method using the above equivalent formulation.

As the name suggested, the semi-smooth Newton method is a Newton-type method. It employs a generalized Jacobian to search for a descent direction of some merit function. Here, by generalized Jacobian, we mean the Clarke generalized Jacobian. Before we present the definition of the generalized Jacobian, we first introduce the concept of a limiting Jacobian, from which the definition of the generalized Jacobian is derived.

Definition 3.3.1 (limiting Jacobian). Let a function $G : \Omega \subseteq \mathbf{R}^m \rightarrow \mathbf{R}^n$, with Ω open, be locally Lipschitz around a neighborhood of $x \in \Omega$, then a limiting Jacobian Jac $G(x)$ of G at x is

$$\text{Jac } G(x) \equiv \{\Upsilon \in \mathbf{R}^{mn} : \Upsilon = \lim_{k\rightarrow+\infty} JG(x_k) \text{ for some sequence } x_k \rightarrow x, x_k \notin N_G\},$$

with N_G being the negligible set of points at which G is not F-differentiable.

Here, a subset $\chi \subseteq \mathbf{R}^m$ is called negligible if for every $\epsilon > 0$, there is a family of boxes $\{\chi_k\}$ with m-dimensional volume $\epsilon_k > 0$ such that

$$\chi \subseteq \bigcup_{k=1}^{\infty}\{\chi_k\}, \text{ and } \sum_{k=1}^{\infty}\epsilon_k < \epsilon.$$

We now present the definition of the Clarke generalized Jacobian based on the limiting Jacobian [FP03a, Definition 7.1.1].

Definition 3.3.2 (Clarke generalized Jacobian). Let a function $G : \Omega \subseteq \mathbf{R}^m \to \mathbf{R}^n$, with Ω open, be locally Lipschitz around a neighborhood of $x \in \Omega$, then the Clarke generalized Jacobian of G at x is

$$\partial G(x) = \text{conv Jac } G(x).$$

Here, conv χ represents the convex hull of the set χ. According to Definition 3.3.2, we know that any limiting Jacobian Jac $G(x)$ belongs to the generalized Jacoobian $\partial G(x)$. In terms of the merit function used in the semi-smooth Newton method, we use the following squared Euclidean norm of $\mathbf{F}_{\text{FB}}$:

$$\theta_{\text{FB}}(z) \equiv \frac{1}{2}\mathbf{F}_{\text{FB}}(z)^T\mathbf{F}_{\text{FB}}(z). \tag{3.3}$$

This merit function (3.3) is continuous differentiable, and its gradient can be calculated by [FP03a, Proposition 9.1.4].

$$\triangledown\theta_{\text{FB}}(z) = H^T\mathbf{F}_{\text{FB}}(z),\ \forall H \in \partial\mathbf{F}_{\text{FB}}(z).$$

We now describe the semi-smooth Newton method in more details. At each iteration k, we search for a direction d^k which could lead to a potential decrease of $\theta_{\text{FB}}(z)$ by solving

$$\mathbf{F}_{\text{FB}}(z^k) + H^k d^k = 0,$$

with the matrix H^k being an element of $\partial\mathbf{F}_{\text{FB}}(z^k)$. Once we determine the direction d^k, we calculate the step size by the well-known Armijo inexact line search rule [A$^+$66]. We find the smallest non-negative integer i_k such that

$$\theta_{\text{FB}}(z^k + 2^{-i_k}d^k) \leq \theta_{\text{FB}}(z^k) + \gamma 2^{-i_k}\nabla\theta_{\text{FB}}(z^k)^T d^k,\ \gamma \in (0, 1),$$

and set the step size as 2^{-i_k}. We repeat the cycle of the generation of a search direction and the determination of the step size until certain stopping criterion is satisfied. In our test, we terminate the algorithm when $\theta_{\text{FB}}(z^k)$ is smaller than a prescribed value. We summarize each step of the semi-smooth Newton method in Algorithm 3. Additionally, we also present a simple procedure to compute a Jac $\mathbf{F}_{\text{FB}}(z)$ in Algorithm 4. This procedure is easy to implement, and we use it to generate $H^k \in \partial\mathbf{F}_{FB}(z^k)$ in Algorithm 3 throughout our experiments.

Notice that in Algorithm 4, e^i is the i-th coordinate vector of length n. Moreover, we have $F(z) = Mz+q$

Algorithm 3: Semi-smooth Newton method

```
**Input**: matrix $M$ of size $n \times n$, vector $q$ of size $n \times 1$, and scalars $\rho > 0$, $p > 0$ and $\gamma \in (0,1)$;
initialize $z^0 \geq 0$, set $k \leftarrow 0$;
**repeat**
    select an element $H^k$ from $\partial \mathbf{F}_{\rm FB}(z^k)$ (Algorithm 4);
    find a solution $d^k$ of the system
        $$\mathbf{F}_{\rm FB}(z^k) + H^k d = 0; \qquad (3.4)$$
    **if** *system* (3.4) *is not solvable, or the condition* $\nabla\theta_{\rm FB}(z^k)^T d^k \leq -\rho\|d^k\|^p$ *is not satisfied,* **then**
        set $d^k \equiv -\nabla\theta_{\rm FB}(z^k)$;
    find the smallest non-negative integer $i_k$ such that
        $$\theta_{\rm FB}(z^k + 2^{-i_k} d^k) \leq \theta_{\rm FB}(z^k) + \gamma 2^{-i_k} \nabla\theta_{\rm FB}(z^k)^T d^k;$$
    set $\tau_k \equiv 2^{-i_k}$;
    set $z^{k+1} \equiv z^k + \tau_k d^k$;
    let $k \leftarrow k+1$;
**until** *a stopping test is satisfied*;
```

Algorithm 4: Procedure to calculate H

```
**Input:** a vector $z \in \mathbf{R}^n$;
set $\beta \equiv \{i : z_i = 0 = F_i(z)\}$;
define $x \in \mathbf{R}^n$ such that $x_i = 1$ for all $i \in \beta$ and $x_i = 0$ for all $i \notin \beta$;
**for** $i = 1, 2, \ldots, n$ **do**
    **if** $i \notin \beta$ **then**
        set the $i$-th column of $H^T$ equal to
        $\left(\frac{z_i}{\sqrt{z_i^2 + F_i(z)^2}} - 1\right) e^i + \left(\frac{F_i(z)}{\sqrt{z_i^2 + F_i(z)^2}} - 1\right) \nabla F_i(z)$;
    **else**
        set the $i$-th column of $H^T$ equal to
        $\left(\frac{x_i}{\sqrt{x_i^2 + (\nabla F_i(z)^T x)^2}} - 1\right) e^i + \left(\frac{\nabla F_i(z)^T x}{\sqrt{x_i^2 + (\nabla F_i(z)^T x)^2}} - 1\right) \nabla F_i(z)$;
```

with M being a tridiagonal matrix for any LCP derived under the jump diffusion models. Therefore, we have $JF(z) = M$. Since the i-th column of the matrix H^T is a linear combination of e^i and $\nabla F_i(z)$, the matrix H^T and H are tridiagonal matrices in our experiments.

3.4 Pivoting method

In Section 3.1–Section 3.3, we have introduced three iterative methods for solving $LCP(q, M)$. In this section, we review a parametric pivoting method designed for solving $LCP(q, M)$ within n steps, where n is the size of the square matrix M. The basic idea of a parametric pivoting method is to augment the $LCP(q, M)$ by incorporating a positive vector p and solving $LCP(q + \theta p, M)$. The vector p is called a parametric vector. Within the pivoting algorithm, the parameter θ is iteratively driven to zero in less than or equal to n steps, which recovers a solution to $LCP(q, M)$.

We now summarize the parametric pivoting method in more details. Denote $L \equiv \{\emptyset\}$ as the index set of the basic z-variables (basic variables in z) and $K = \{1, 2, \dots, n\}$ as the index set of the non-basic z-variables (non-basic variables in z), with basic variables and non-basic variables defined by

Definition 3.4.1 (Basic and Non-Basic Variables). For any $LCP(q, M)$ and $z \in SOL(q, M)$, define $w = Mz + q$. Consider the following system of linear equality constraints and bound constraints:

$$Ax = -q \text{ and } x \geq 0$$

with $A = [-I \; M]$ and $x^T = [w^T, z^T]$. Any set $B \subset \{1, 2, \dots, 2n\}$ such that $|B| = n$ and rank $A_{\cdot,B} = n$ is called a basis, and x_B are called the basic variables. The complementary set of the basic variables are called the non-basic variables.

We initialize the vector $\bar{q} = q$ and the parametric vector $\bar{p} = p$. At each iteration, we apply a pivot to the $LCP(\bar{q} + \theta\bar{p}, M)$ and determine the next pivot by performing the following ratio test:

$$\theta = \max\{-\bar{q}_i/\bar{p}_i,\ i \in K \text{ and } \bar{p}_i > 0\}.$$

Next, we increase the index set of the basic z-variables L by adding an index $k \in K$ that maximizes θ, and update the parametric vector $\bar{q}$ and $\bar{p}$ using equation

$$(\bar{q}_{K'}, \bar{p}_{K'})_{\text{new}} = (\bar{q}_{K'}, \bar{p}_{K'})_{\text{old}} - (M_{K'k} - M_{K'L}\bar{M}_{Lk})(\bar{q}_k, \bar{p}_k)_{\text{old}}/(M_{kk} - M_{kL}\bar{M}_{Lk}). \tag{3.5}$$

Here, $K' = K \setminus \{k\}$ is the new index set for the non-basic z-variables, and $\bar{M}_{Lk}$ is the solution to the system of linear equations

$$M_{LL}\bar{M}_{Lk} = M_{Lk}. \tag{3.6}$$

We then update the index set of the non-basic z-variables K by K'. We repeat the above procedure until either $\bar{p}_K \leq 0$ or $\theta \leq 0$, and we have found the index set of the basic z-variables of the original $LCP(q, M)$. A detailed description of the parametric pivoting method is summarized in Algorithm 5.

Algorithm 5: Parametric principal pivoting method

```
**Input**: Matrix $M$ of size $n \times n$, vector $q$ of size $n \times 1$, parametric vector $p > 0$ of size $n \times 1$;
initialize $L = \{\emptyset\}$ and $K = \{1, \ldots, n\}$;
set $\bar{p} = p$, $\bar{q} = q$;
repeat
    determine the critical value $\bar{\theta} = \max\{-\bar{q}_i/\bar{p}_i : i \in K \text{ and } \bar{p}_i > 0\}$;
    if $\bar{p}_K \leq 0$ or $\bar{\theta} \leq 0$ then
        break;
    select $k \in K$ to be a maximizing index;
    repeat
        if $K = \{k\}$ then
            set $K = \{\emptyset\}$ and $L = \{1, \ldots, n\}$;
            break;
        else
            let $K' = K \setminus \{k\}$ and $L = L \cup \{k\}$ ;
            solve the system of linear equations below for $\bar{M}_{Lk}$:
                $$M_{LL}\bar{M}_{Lk} = M_{Lk}; \quad (3.7)$$
            update the vector $\bar{q}$ and the parametric vector $\bar{p}$:
                $$(\bar{q}_{K'}, \bar{p}_{K'})_{\text{new}} = (\bar{q}_{K'}, \bar{p}_{K'})_{\text{old}} - (M_{K'k} - M_{K'L}\bar{M}_{Lk})(\bar{q}_k, \bar{p}_k)_{\text{old}}/(M_{kk} - M_{kL}\bar{M}_{Lk});$$
            update the index set of the non-basic $z$-variables $K = K'$;
    until;
until;
solve the system of linear equations for $z^*_L$: $M_{LL}z^*_L = -q_L$;
the vector $z = (z^*_L, 0)$ is the unique solution to the $LCP(q, M)$.
```

The theorem below guarantees that under certain conditions, Algorithm 5 terminates in at most n steps.

Theorem 3.4.2. *Let M be a P-matrix of order n and q a vector in $\mathbf{R}^n$. If the parametric vector p satisfies*

$$M^{-1}_{LL}p_L \geq 0, \ \forall L \subseteq \{1, 2, \ldots, n\}, \tag{3.8}$$

then the parametric principal pivoting algorithm (Algorithm 5) will compute a solution to $LCP(q, M)$ in at most n pivots.

The proof of Theorem 3.4.2 is based on the observation that anytime a non-basic variable becomes basic during a pivot, it will remain basic for the remainder of the algorithm. For the augmented $LCP(q+\theta p, M)$, all the variables of z initially are non-basic. At each pivot, we increase the size of the basic set by one. Therefore, the method will terminate after at most n steps. The proof of Theorem 3.4.2 can be found in [PC85].

The success of the parametric pivoting method depends on the ability to find a parametric vector $p > 0$ such that condition (3.8) holds. In our numerical test, we let $p^T = (1, \ldots, 1)$. Given our choice of the parametric vector p, we are able to solve LCPs derived from American option pricing problems. In general, for a strictly diagonally dominant matrix $M \in \mathbf{R}^{n\times n}$ and its comparison matrix $\bar{M} \in \mathbf{R}^{n\times n}$ defined by

$$\bar{M}_{ij} = \begin{cases} |M_{ij}|, & \text{if } i = j, \\ -|M_{ij}|, & \text{if } i \neq j, \end{cases}$$

if we can find a vector $\bar{p} > 0$ such that $\bar{M}\bar{p} > 0$, then the vector $p = \left(\frac{M+\bar{M}}{2}\right)\bar{p}$ satisfies condition (3.8) [PC85, Corollary 1].

By assuming the existence of a vector p such that condition (3.8) holds, we can analyze the complexity of the parametric pivoting method based on Algorithm 5. At each iteration, we require $O(n)$ steps to determine the value of θ and update the index set of the basic (non-basic) z-variables L (K). We then solve the system of linear equations (3.6), which requires a time complexity of $O(n)$. This is because we employ the Gaussian elimination method in our implementation, which solves system (3.6) with a tridiagonal matrix M_{LL} in $O(n)$ steps (see [GVL12]). Therefore, we update the vector $\bar{q}$ and $\bar{p}$ in $O(n^2)$. Since we are guaranteed to terminate the algorithm in at most n steps, the total complexity of applying the parametric pivoting method to solve the $LCP(q, M)$ is $O(n^3)$.

3.5 Numerical comparison of four LCP solvers

In this section, we report the numerical performance of the above four methods—the PSOR method, a two phase active-set method, a semi-smooth Newton method and a pivoting method—for solving option pricing problems under Kou's jump diffusion model and the Dupire system. The experimental codes of the four algorithms are written in MATLAB for a fair evaluation of each algorithm. Before presenting the computational results, we first give some details involved in the implementation of the four algorithms. For the PSOR method, the two phase active-set method and semi-smooth Newton method, an initial guess of the LCP's solution at each time step is required. Consider the problem of solving a sequence of $LCP(q^j, M)$ for

$j = 1, 2, \ldots, m$. We solve the first $LCP(q^1, M)$ by setting the initial iterate to be zero, and use the solution z^{j-1} of the $LCP(q^{j-1}, M)$ as the initial iterate in solving $LCP(q^j, M)$, $j = 2, 3, \ldots, m$. In terms of the implementation of the PSOR method and the two phase active-set method, we terminate both algorithms when either

$$\| \min(z^j, Mz^j + q^j) \|_\infty \leq \epsilon_{\text{sor}}, \tag{3.9}$$

with ϵ_{sor} being the tolerance level of the error or a maximum number of 1000 iterations has been reached. If both methods are terminated after 1000 iterations without satisfying condition (3.9), the 1000-th iterate will be deemed satisfactory and used as the solution of the corresponding LCP. Following [FLMN10], we choose the relaxation parameters of both methods as

$$\omega = \frac{2}{1 + \sqrt{1 - \rho^2}}, \text{ with } \rho = \max_i \frac{1}{M_{ii}} \sum_{j \neq i} \mid M_{ij} \mid .$$

Notice that in the reduced-space phase of the two phase active-set method, a few systems of linear equations need to be solved at each iteration. In the case of a tridiagonal matrix M, we apply the Gaussian elimination method to solve the corresponding systems of linear equations. In terms of the implementation of the semi-smooth Newton method, we have shown in Section 3.3 that $H^k \in \partial \mathbf{F}_{\text{FB}}(z^k)$ at any iteration k is a tridiagonal matrix. Therefore, we can again employ the Gaussian elimination method to solve equation (3.4). Since $\mathbf{F}_{\text{FB}}(z) = 0$ implies $z \in SOL(q, M)$, the norm of the vector $\mathbf{F}_{\text{FB}}(z)$ can be used to measure the accuracy of the computed solution. We use

$$|| \mathbf{F}_{\text{FB}}(z) ||_\infty \leq \epsilon_{\text{semi}}$$

as the stopping criterion of the semi-smooth Newton method. As for the implementation of the parametric pivoting method, we set the parametric vector p to be a vector with all entries one throughout all our experiments. In the pivoting method, Equation (3.7) is again solved by the Gaussian elimination method due to the tridiagonal structure of the matrix M. Now we have presented the numerical techniques involved in the implementation of the four algorithms, we report the numerical performance of the algorithms in the following section.

3.5.1 Tests with Kou's jump diffusion model

In this part, we consider pricing American put options via solving the complementarity system (2.13) derived under Kous' jump diffusion model. In our test, we price one year ($T = 1$) American put options with a fixed strike price $K = 100$. Following the assumptions of Kou's jump diffusion model, we set the risk-free interest rate of the market, the dividend and the volatility of the underlying asset in the model as the following constants:

$$r = 0.05, \ \ q = 0, \ \ \sigma = 0.1.$$

We also fix the jump parameters of the model at

$$\lambda = 3, \ \ p = 0.2, \ \ \eta_1 = 40, \ \ \eta_2 = 12.$$

The computational domain of the log of the spot price x is set as follows for the computational purpose.

$$\underline{x} = -1, \ \ \bar{x} = 1.$$

To achieve comparable results, we let

$$\epsilon_{\text{sor}} = \epsilon_{\text{semi}} = 10^{-8}$$

as the error tolerance level for the PSOR method, the two phase active-set method and the semi-smooth Newton method. In addition, for the semi-smooth Newton method, we set the scalar parameter γ as 0.8. For the two phase active-set method, we set $k_{\text{sor}} = 3$ and $\Delta_{\text{ac}} = 20$. Table 3.1 summarizes the results of the four methods applied to LCPs obtained under Kou's jump diffusion model with space discretization $R = 400$ and 800, respectively. In Table 3.1, we list the time discretization Q in its first column and report the total computing time in seconds (CPU) for all four methods. We report the average number of iterations used by the PSOR method, the two phase active-set method and the semi-smooth Newton method to solve the sequence of LCPs as Iter. This number reflects the average performance of the above three algorithm for solving all LCPs. Moreover, for the two phase active-set method and the semi-smooth Newton method, the average number of the reduced-space iterations (Red iter) and the average total number of the line search iterations (Line iter) per LCP are presented in Table 3.1 as well for a thorough comparison of the algorithms. Table 3.2 summarizes the corresponding pricing errors of the four methods with different discretizations. To calculate the pricing error of an approximate solution, we create the error vector by taking the difference between the calculated option prices and the benchmark prices at five points:

$$x \in \{-0.02, \ -0.01, \ 0, \ 0.01, \ 0.02\}. \tag{3.10}$$

The pricing error is then calculated as the maximum norm of the error vector. The five points are chosen such that they correspond to the five asset prices that are in the domain of interest. The benchmark prices

	PSOR	Iter	Active-set	Iter	Red iter	Newton	Iter	Line iter	Pivot
	R=400								
Q=50	0.0370	34.56	0.0177	7.00	1.00	0.2054	5.08	7.38	1.3873
Q=100	0.0508	23.70	0.0194	4.69	1.00	0.2766	3.68	5.65	2.7916
Q=500	0.1013	9.53	0.0770	4.01	1.00	0.5751	1.51	1.912	13.6247
	R=800								
Q=50	0.1438	65.34	0.0436	14.62	1.00	0.4529	7.34	12.02	5.3950
Q=100	0.1949	45.00	0.0471	8.14	1.00	0.5619	4.71	7.28	9.6179
Q=500	0.3979	18.64	0.1379	4.20	1.00	1.1370	1.87	2.83	48.3319

Table 3.1: Kou's jump diffusion model: computational time for $R = 400, 800$.

of the corresponding put options are computed with sufficiently large discretizations of time and space. In our experiments, the benchmark prices associated with the set x given by (3.10) are

$$\{5.2570,\ 5.8447,\ 6.4607,\ 7.1043,\ 7.7754\}.$$

As shown in Table 3.1 and Table 3.2, the two phase active-set method reaches the same accuracy as the other three methods within a significantly shorter period of time. The average number of iterations (Iter) further suggests that the two phase active-set method greatly reduces the number of PSOR iterations, and hence accelerates the convergence. Additionally, we notice that the pivoting method takes substantially more time than the other three methods to find an approximate solution of the LCP. Therefore, we only test the PSOR method, the two phase active-set method and the semi-smooth Newton method when the space discretization is $R = 1600$. We report the corresponding performance of the three algorithms in Table 3.3 along with their pricing errors in Table 3.4. The computational results of Table 3.3 and Table 3.4 are consistent with the results of Table 3.1 and Table 3.2. The two phase active-set method outperforms both the PSOR method and the semi-smooth Newton method with different space discretizations. Also, the performance of the PSOR method and the semi-smooth method tend to be more similar to each other as we use a finer time discretization.

3.5.2 Tests with the Dupire system

In this part, we consider pricing problems of American put options under the Dupire system (2.19). As suggested by the results of Table 3.1 and Table 3.2, the pivoting method does not work particularly well in terms of solving LCPs with tridiagonal matrices derived under the Kou's jump diffusion model. Therefore, we only compare the numerical performance of the PSOR method, the two phase active-set method and the semi-smooth Newton method in solving LCPs derived under the Dupire system. We again consider pricing

	PSOR	Active-set	Newton	Pivot
	R=400			
Q=50	0.1775	0.1775	0.1775	0.1775
Q=100	0.0902	0.0902	0.0903	0.0902
Q=500	0.0178	0.0178	0.0182	0.0178
	R=800			
Q=50	0.1779	0.1779	0.1779	0.1779
Q=100	0.0907	0.0907	0.0908	0.0907
Q=500	0.0183	0.0183	0.0191	0.0183

Table 3.2: Kou's jump diffusion model: error for $R = 400, 800$.

	PSOR	Iter	Active-set	Iter	Red iter	Newton	Iter	Line iter
	R=1600							
Q=50	0.5267	123.86	0.2204	27.46	1.00	1.2430	11.70	23.02
Q=100	0.7245	84.94	0.1617	14.47	1.00	1.5653	7.35	12.31
Q=500	1.4870	35.08	0.2817	4.74	1.00	2.8310	2.67	4.438
Q=1000	2.0626	24.02	0.5161	4.27	1.00	3.9154	1.85	2.865

Table 3.3: Kou's jump diffusion model: computational time for $R = 1600$.

	PSOR	Active-set	Semi
	R=1600		
Q=50	0.1780	0.1780	0.1780
Q=100	0.0908	0.0908	0.0908
Q=500	0.0184	0.0184	0.0184
Q=1000	0.0091	0.0091	0.0091

Table 3.4: Kou's jump diffusion model: error for $R = 1600$.

one year ($T = 1$) American put options with spot price $S = 100$ in this experiment. The risk-free interest rate of the market, the dividend and the volatility of the underlying asset used in this experiment are set as

$$r = 0.05, \;\; q = 0, \;\; \sigma = 0.1.$$

We also fix the jump parameters of the model at

$$\lambda = 3, \;\; p = 0.2, \;\; \eta_1 = 40, \;\; \eta_2 = 12.$$

We restrict the log of the strike price x on the finite computational domain

$$\underline{x} = -1, \;\; \bar{x} = 1$$

for the computational purpose. To achieve comparable results of the three algorithms, we again use

$$\epsilon_{\text{sor}} = \epsilon_{\text{semi}} = 10^{-8}$$

as the error tolerance levels for the PSOR method, the active-set method and the semi-smooth Newton method. In addition, we set the scalar parameter γ to be 0.8 for the semi-smooth Newton method and $k_{\text{sor}} = 3$ and $\Delta_{\text{ac}} = 20$ for the two phase active-set method.

Table 3.5 summarizes the numerical results of the three methods, the PSOR method, the two phase active-set method and the semi-smooth Newton method for solving the Dupire system with different space discretizations: $Q = 400, \; 800, \; 1600$. Specifically, in Table 3.5, we list the number of time discretization Q in its first column and report both the total computing time in seconds (CPU) and the number of average iterations (Iter) for all three methods. In addition, we report the average number of reduced-space iterations (Red iter) for the two phase active-set method and the average total number of the line search iterations (Line iter) for the semi-smooth Newton method. In Table 3.6, we report the corresponding pricing errors of the three methods. The errors are calculated in the same way discussed in Section 3.5.1, and the corresponding error vectors are calculated by taking the difference of the calculated option prices and the theoretical option prices at five points:

$$x \in \{-0.02, \; -0.01, \; 0, \; 0.01, \; 0.02\}. \tag{3.11}$$

The benchmark values of the corresponding put options are computed with sufficiently large discretizations

	PSOR	Iter	Active-set	Iter	Red iter	Newton	Iter	Line iter
	R=400							
Q=50	0.0279	25.98	0.0133	4.00	1.00	0.2051	5.06	7.42
Q=100	0.0421	19.32	0.0173	4.00	1.00	0.2821	3.67	5.63
Q=500	0.0901	8.32	0.0775	4.00	1.00	0.5852	1.53	1.94
	R=800							
Q=50	0.1145	49.48	0.0192	4.00	1.00	0.454	7.38	12.10
Q=100	0.1587	36.60	0.0284	4.00	1.00	0.5644	4.70	7.25
Q=500	0.3624	16.88	0.1330	4.00	1.00	1.1649	1.85	2.80
	R=1600							
Q=50	0.4051	93.76	0.0355	4.78	1.00	1.3498	11.68	23.18
Q=100	0.607	69.19	0.0519	4.00	1.00	1.6299	7.40	12.47
Q=500	1.3624	31.91	0.2448	4.00	1.00	2.8977	2.67	4.43
Q=1000	1.9406	22.52	0.4916	4.00	1.00	3.9947	1.85	2.86

Table 3.5: The Dupire system: computational time for $R = 400, 800, 1600$.

in time and space. In our experiments, the benchmark values associated with the set x given by (3.11) are

$$\{5.64144,\ 6.03878,\ 6.46078,\ 5.90360,\ 5.36330\}.$$

Observing the results from Table 3.5 and Table 3.6, we note that the two phase active-set method reaches the same accuracy as the other two methods within a significantly shorter time in all tests. Moreover, the average number of iterations (Iter) in Table 3.5 further confirms that the two phase active-set method does greatly reduce the number of PSOR iterations, and hence accelerates convergence. Different from the results shown in Table 3.1 and Table 3.3, the two phase active-set method on average requires a smaller number of iterations (Iter) to solve the LCPs of this experiment compared to the PSOR method and the semi-smooth Newton method.

	PSOR	Active-set	Newton
	R=400		
Q=50	0.1795	0.1795	0.1795
Q=100	0.0936	0.0936	0.0936
Q=500	0.0226	0.0226	0.0226
	R=800		
Q=50	0.1768	0.1768	0.1768
Q=100	0.0907	0.0907	0.0907
Q=500	0.0193	0.0193	0.0193
	R=1600		
Q=50	0.1761	0.1761	0.1761
Q=100	0.0900	0.0900	0.0900
Q=500	0.0185	0.0185	0.0185
Q=1000	0.0094	0.0094	0.0094

Table 3.6: The Dupire system: error for $R = 400, 800, 1600$.

Chapter 4

Model calibration with American option data

For any model, the estimation of the parameters is of vital importance to the accurate evaluation of option prices. For example, an important input in the analysis of the BSM model is the volatility parameter σ of the underlying asset. For an European call option, its price can be explicitly expressed by the following formula under the BSM model. Details of the derivation can be found in [BS73].

$$c(S,t) = SN(d_1) - Ke^{-r(T-t)}N(d_2). \tag{4.1}$$

Similarly, for an European put option, its price can be calculated by

$$p(S,t) = Ke^{-r(T-t)}N(-d_2) - SN(-d_1). \tag{4.2}$$

Here, d_1 and d_2 are two constants containing the volatility parameter σ. Therefore, given a call option of value c or a put option of value p, the spot price S of the underlying asset, the strike price K, the maturity T of the option and the current time t, we can calculate the volatility implied by the option price via solving equations (4.1) or (4.2). The calculated volatility is called the implied volatility. Since the implied volatility is a complicated function of the above mentioned factors, we can employ a Newton method to solve the equation numerically. In practice, traders assume the implied volatility of an option as a function of its strike price known as a volatility smile for a given expiration. They then plot the implied volatilities of the options with different maturities and use the volatility function as a pricing tool to determine the prices of exotic options.

However, not all the option prices can be expressed explicitly under different models. For instance, in case of American options, there is no explicit formula or finite procedure for computing the exact prices under the BSM model. Therefore, the development of a fast and accurate calibration procedure of the parameters using the available option data from the market is of great practical value. Different approaches for dealing with the unknown parameters have been proposed by researchers. Among them are statistical estimation methods based on historical data. In [CFH93], Robert, Steven and Joel suggested using an Egarch model to forecast the volatilities and their correlations. Another popular approach for calibrating the unknown parameters is via mathematical modeling of the parameters. Heston in [Hes93] suggested using two-dimensional stochastic differential equations to describe both the asset price process S_t and the variance process $v_t = \sigma_t^2$. Derman and Kani in [DK94], Dupire in [Dup94] and Rubinstein in [Rub94] developed the implied volatility function (IVF), or the implied tree model for the estimation of the volatility parameter. This special class of models provides close fit to the prices of plain vanilla options and helps traders eliminating the arbitrages arising form the banks' internal models. In addition to the above calibration methods, Huang and Pang in [HP00] established an optimization approach, which searched for an volatility function by constraining on the American options. In this thesis, we mainly focus on estimating the volatility parameter under the local volatility model via the optimization approach.

As discussed in Section 2.2.3, upon the discretization of the partial differential inequalities that are satisfied by American options under the local volatility model, we obtain linear complementarity problems. With properly chosen discretization steps, the solutions of the LCPs provide close approximations to the values of American options. Therefore, the problem of computing a volatility parameter of American options can be constructed as an inverse problem of the American option pricing problem. More precisely, it belongs to a class of constrained optimization problems with finite-dimensional linear complementarity systems as part of the constraints. In addition, we take into account the available market data by minimizing the difference between the theoretical prices and the given market prices in this type of optimization problems. Hence, both the volatility parameter and the theoretical prices are unknowns in the optimization problem.

Once we reformulate our calibration problem, it becomes an instance of mathematical programs with complementarity constraints (MPCCs). Luo, Pang and Ralph provided a comprehensive study of this class of constrained optimization problems in [LPR96]. Due to the presence of the complementarity constraints, the well-known Karush-Kuhn-Tucker (KKT) optimality conditions no longer hold for MPCCs. In [SS00], Scheel and Scholtes studied the stationarity concepts of MPCCs, based on a piece-wise smooth formulation. The discontinuity and the non-smoothness of the MPCC also motivate novel approaches to solve this class of problems. In [FLP98], Fukushima, Luo and Pang proposed a globally convergent sequential quadratic programming algorithm for solving MPCCs. Leyffer, Lopez-Calva and Nocedal in [LLCN06] suggested using interior methods for solving MPCCs. In our work, we follow a similar approach, an implicit programming algorithm (IMPA) proposed by Huang and Pang in [HP00] to solve the MPCC derived under the local volatility model.

However, instead of applying IMPA directly to our inverse pricing problem, a few modifications are made both in terms of the construction of the inverse problem and the implementation of IMPA itself. Unlike modeling the volatility function by a two-dimensional mesh in [HP00], we construct a smooth local volatility function by the linear combination of some basis functions. This approach is inspired by Coleman, Li and Verma, who in [CLV99] used a two-dimensional spline approximation to construct the local volatility function for European options. Also, we modify IMPA by the further decomposition of the associated matrices of the complementarity constraints in the MPCCs. The decomposition of the matrices not only allows the modified IMPA to handle MPCCs of large sizes, but also improves the numerical efficiency of the modified IMPA by taking advantage of the sparsity of the associated matrices.

In addition to the modification of IMPA, we propose a new hybrid algorithm with improved performance for solving MPCCs. This new hybrid algorithm, combining the modified IMPA along with a grid search method, demonstrates the capability of reproducing option prices that closely match the observed prices within a reasonable amount of time.

The remainder of this chapter is organized as follows. In Section 4.1, we review the definition of the MPCC. In Section 4.2, we present the formulation of the inverse pricing problems as MPCCs under two different models, the Dupire system (2.19) and the local volatility model (2.11). We then proceed with the presentation of the modified IMPA and the new hybrid algorithm for solving MPCCs with tridiagonal matrices in Section 4.3. In the last section of this chapter, we compare the numerical performance of the modified IMPA with a grid search method for solving the calibration problems under the Dupire system. We also test the modified IMPA and the hybrid algorithm with the calibration problems formulated under the local volatility model with synthetic and real market data.

4.1 Mathematical programs with complementarity constraints

Before we present the formulation of the calibration problem of the volatility parameter for American options as an MPCC, we first review the definition of an MPCC. In general, for a given set $\mathbf{\Gamma}$ and an objective function θ, an MPCC is defined by

$$\begin{array}{ll} \text{minimize} & \theta(\sigma, v) \\ \text{subject to} & \sigma \in \Gamma, \\ \text{and} & 0 \leq v - p \perp q(\sigma) + M(\sigma)v \geq 0. \end{array}$$

Moreover, if v can be viewed as a function of σ (i.e., the American option price can be viewed as a function of its volatility under the BSM model), we can further rewrite the above optimization problem as an implicit program in the variable of σ.

$$\begin{aligned} &\text{minimize} && \phi(\sigma) \equiv \theta\left(\sigma, v(\sigma)\right) \\ &\text{subject to} && \sigma \in \Gamma. \end{aligned}$$

In the next two subsections, we present the formulation of the calibration problems of the volatility parameter as MPCCs under the Dupire system and the local volatility model. Along with the formulation of the calibration problems, we also provide a detailed description of the construction of the volatility function via the two-dimensional basis functions.

4.2 Formulation of the calibration problems as MPCCs

This section presents the formulation of the calibration problems of an implied volatility parameter of American options as MPCCs. We build our formulation on the linear complementarity systems (2.19) and (2.11) satisfied by the American options under the Dupire system and the local volatility model. It is worth mentioning that though throughout this work, we only consider the calibration problems of American options, the same approach can be applied to the calibration problems of exotic options as well as other financial derivatives under different models.

4.2.1 The MPCC of the local volatility model

The basic framework of the local volatility model is that for an underlying asset, its spot price S_t follows the stochastic differential equation described in equation (2.7). Notice that the volatility $\sigma(S_t, t)$ is an unknown function of the spot price S_t and the current time t. Our goal here is to approximate the local volatility function $\sigma(S_t, t)$ by solving an optimization problem with linear complementarity constraints arising form the characteristic of American options.

In our study, we approximate the local volatility function by a linear combination of the two-dimensional basis functions. As mentioned in Section 2.2.3, the coefficients of the one-dimensional basis functions in the Dupire system are the time values of American options with different strike prices. Similarly, the coefficients of the basis functions here represent the volatility values of the grid points on the mesh used for the construction of the basis functions and are to be determined by solving the corresponding MPCC. We now introduce the construction of the local volatility function via a linear combination of the two-dimensional basis functions, followed by the formulation of the corresponding MPCC.

We again restrict our calibration problem on a finite domain $\Omega = [0, S_{\max}] \times [0, T]$, and choose a set of evenly distributed grid points $\{(\bar{s_g}, \bar{t_k})\}$, $g = 1, 2, \ldots, p_1$ and $k = 1, 2, \ldots, p_2$ on the finite domain Ω. Let h_s

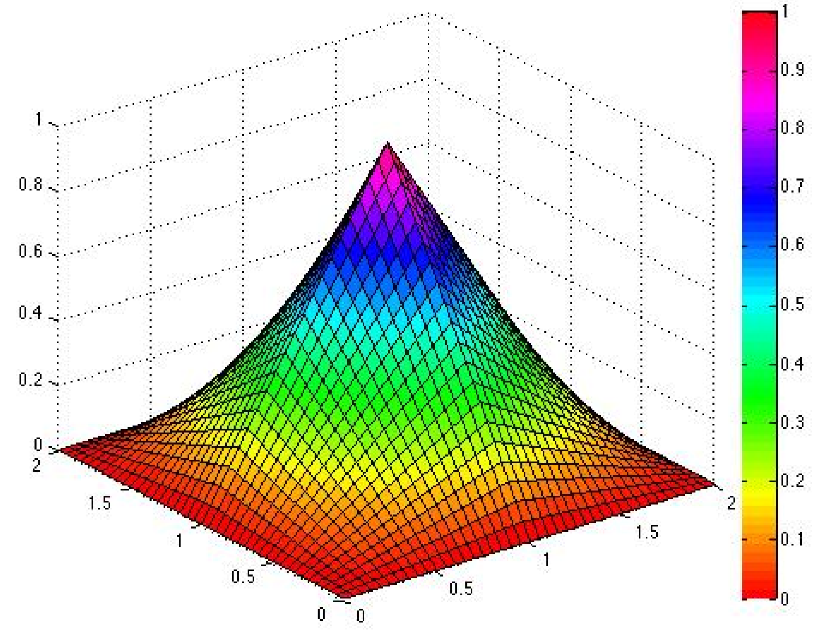

Figure 4.1: $\phi_{1,1}(s,t)$ for $s_1 = 1$, $t_1 = 1$ and $h_s = h_t = 1$.

and h_t be the discretization steps of the time and space variables. We have

$$h_s = S_{\max}/(p_1 - 1), \text{ and } h_t = T/(p_2 - 1).$$

Given the above discretization of the domain Ω, we define the two-dimensional basis functions as follows.

$$\begin{aligned}\phi_{gk}(s,t) =& \phi_{h_s,g}(s)\phi_{h_t,k}(t) \\ =& \phi_{1,0}\left((s - \bar{s}_g)/h_s\right)\phi_{1,0}\left((t - \bar{t}_k)/h_t\right),\end{aligned}$$

for $g = 1, 2, \ldots, p_1$ and $k = 1, 2, \ldots, p_2$. According to the above definition, we notice that a two-dimensional basis function is the product of two one-dimensional basis functions. To provide readers a better view of the two-dimensional basis functions, we plot one two-dimensional basis function with $s_g = t_k = 1$ and $h_s = h_t = 1$ in Figure 4.1. This basis function has value zero outside the range $(0,2) \times (0,2)$ and it reaches its peak value one at point $(1,1)$.

According to our construction of the volatility function, the coefficients of the basis functions represent the volatility values at the grid points $\{(s_g, t_k)\}$, $g = 1, 2, \ldots, p_1$ and $k = 1, 2, \ldots, p_2$. We denote the volatility value at the grid point (s_g, t_k) as $\bar{\sigma}_{gk}$. Hence, the local volatility $\sigma(S_t, t)$ can be approximated by the linear combination of the above basis functions as follows.

$$\sigma(s,t) = \sum_{g,k} \bar{\sigma}_{gk} \phi_{gk}(s,t).$$

Let $\bar{\sigma} \equiv (\bar{\sigma}_{11}, \dots, \bar{\sigma}_{p_1 p_2})^T$. By our construction of the volatility function $\sigma(s,t)$, we can rewrite it as a function of the spot price S, the current time t and the volatility parameter $\bar{\sigma}$: $f(s,t,\bar{\sigma})$. Denote

$$\sigma_i^m \equiv f(i\delta S, m\delta t, \bar{\sigma}).$$

Apply the above notation to the complementarity system (2.24) obtained under the local volatility model via the finite difference method. The entries of the matrices $\mathbf{M}^m$ and $\widehat{\mathbf{M}}^m$ of time $t = m\delta t$ can be updated as follows.

$$\begin{aligned}
a_i &\equiv -\frac{1}{2}\theta f(i\delta S, m\delta t, \bar{\sigma})^2 i^2 + \frac{\theta(r-q)i}{2}, \\
b_i &\equiv \frac{1}{\delta t} + \theta f(i\delta S, m\delta t, \bar{\sigma})^2 i^2 + r, \\
c_i &\equiv -\frac{1}{2}\theta f(i\delta S, m\delta t, \bar{\sigma})^2 i^2 - \frac{\theta(r-q)i}{2}, \\
\hat{a}_i &\equiv -\frac{1}{2}(1-\theta) f(i\delta S, m\delta t, \bar{\sigma})^2 i^2 + \frac{(1-\theta)(r-q)i}{2}, \\
\hat{b}_i &\equiv -\frac{1}{\delta t} + (1-\theta) f(i\delta S, m\delta t, \bar{\sigma})^2 i^2, \\
\hat{c}_i &\equiv -\frac{1}{2}(1-\theta) f(i\delta S, m\delta t, \bar{\sigma})^2 i^2 - \frac{(1-\theta)(r-q)i}{2}.
\end{aligned} \tag{4.3}$$

Thus given the time discretization R and the space discretization Q, the matrices $\mathbf{M}^m$ and $\widehat{\mathbf{M}}^m$ at each time $t = m\delta t$ can be viewed as a function of $\bar{\sigma}$ and m. Denote $\mathbf{M}^m$ and $\widehat{\mathbf{M}}^m$ as $\mathbf{M}(m,\bar{\sigma})$ and $\widehat{\mathbf{M}}(m,\bar{\sigma})$. By assuming that condition (2.25) is satisfied by our discretization, we again have the matrix $\mathbf{M}(m,\bar{\sigma})$ being a strictly row diagonally dominant matrix irrespective of the value of $\bar{\sigma}$. Recall that $\mathbf{q}^{m+1} = \widehat{\mathbf{M}}^m \mathbf{V}^{m+1}$ in the local volatility model. Accordingly, we denote $\mathbf{q}^{m+1}$ as $\mathbf{q}(m+1,\bar{\sigma}) = \widehat{\mathbf{M}}(m,\bar{\sigma})\mathbf{V}^{m+1}$. Similarly, we denote $\mathbf{d}^m$ arising from the boundary conditions as $\mathbf{d}(m,\bar{\sigma})$. Hence, the prices of American options satisfy the following sequence of LCPs. For $m = 0, 1, \dots, R-1$,

$$0 \le \mathbf{V}^m - \mathbf{\Psi}^m \perp \mathbf{d}(m,\bar{\sigma}) + \mathbf{q}(m+1,\bar{\sigma}) + \mathbf{M}(m,\bar{\sigma})\mathbf{V}^m \ge 0.$$

We combine the above LCPs together into the following aggregate LCP of size $R(Q-1)$:

$$0 \le \mathbf{V} - \mathbf{\Psi} \perp \mathbf{d}(\bar{\sigma}) + \mathbf{M}_T(\bar{\sigma})\mathbf{V} \ge 0, \tag{4.4}$$

with

$$\mathbf{V} \equiv \begin{pmatrix} \mathbf{V}^0 \\ \vdots \\ \mathbf{V}^{R-1} \end{pmatrix}, \quad \boldsymbol{\Psi} \equiv \begin{pmatrix} \boldsymbol{\Psi}^0 \\ \vdots \\ \boldsymbol{\Psi}^{R-1} \end{pmatrix}, \quad \mathbf{d} \equiv \begin{pmatrix} \mathbf{d}(0,\bar{\sigma}) \\ \vdots \\ \mathbf{d}(R-1,\bar{\sigma}) \end{pmatrix}.$$

Here, the aggregated vector $\mathbf{V}$ contains all option prices, the aggregated vector $\boldsymbol{\Psi}$ contains all payoffs and the vector $\mathbf{d}$ consists all boundary conditions. All three vectors are of size $R(Q-1)$. The matrix $\mathbf{M}_T(\bar{\sigma})$ is a $R(Q-1) \times R(Q-1)$ matrix defined by

$$\mathbf{M}_T(\bar{\sigma}) \equiv \begin{pmatrix} \mathbf{M}(0,\bar{\sigma}) & \widehat{\mathbf{M}}(0,\bar{\sigma}) & & & \\ & \mathbf{M}(1,\bar{\sigma}) & \widehat{\mathbf{M}}(1,\bar{\sigma}) & & \\ & & \ddots & \ddots & \\ & & & \mathbf{M}(R-2,\bar{\sigma}) & \widehat{\mathbf{M}}(R-2,\bar{\sigma}) \\ & & & & \mathbf{M}(R-1,\bar{\sigma}) \end{pmatrix}.$$

Notice that the aggregated matrix $\mathbf{M}_T(\bar{\sigma})$ is a block upper triangular matrix with strictly row diagonally dominant blocks, and hence it is a P-matrix. The P-property of the aggregated matrix $\mathbf{M}_T(\bar{\sigma})$ plays an important role in the design of IMPA, which will be discussed in Section 4.3.

The above formulation can be easily extended to the case of multiple options. Specifically, we embed the LCPs (4.4) for American options with different strike prices together into one large complementarity system. Assume there are L American options with different strike prices, each satisfying the following LCP:

$$0 \leq \mathbf{V}^l - \boldsymbol{\Psi}^l \perp \mathbf{d}(\bar{\sigma}) + \mathbf{M}_T(\bar{\sigma})\mathbf{V}^l \geq 0.$$

We concatenate the L LCPs together and define $\mathbf{M}_{\text{MUL}}(\bar{\sigma})$ as a block diagonal matrix with same block entries $\mathbf{M}_T(\bar{\sigma})$. Then the final LCP formulation for multiple American options on a single underlying asset is expressed as follows.

$$0 \leq \mathbf{V}_{\text{MUL}} - \boldsymbol{\Psi}_{\text{MUL}} \perp \mathbf{d}_{\text{MUL}}(\bar{\sigma}) + \mathbf{M}_{\text{MUL}}(\bar{\sigma})\mathbf{V}_{\text{MUL}} \geq 0, \tag{4.5}$$

where

$$\mathbf{V}_{\text{MUL}} \equiv (\mathbf{V}^l)_{l=1}^L, \ \boldsymbol{\Psi}_{\text{MUL}} \equiv (\boldsymbol{\Psi}^l)_{l=1}^L, \text{ and } \mathbf{d}_{\text{MUL}}(\bar{\sigma}) \equiv (\mathbf{d}(\bar{\sigma}))_{l=1}^L$$

are the aggregated option values, payoffs and boundary conditions. The aggregated matrix $\mathbf{M}_{\text{MUL}}(\bar{\sigma})$ is a $LR(Q-1) \times LR(Q-1)$ block diagonal matrix. Notice that both vectors $\mathbf{d}_{\text{MUL}}(\bar{\sigma})$ and $\mathbf{M}_{\text{MUL}}(\bar{\sigma})$ are functions of the volatility parameter $\bar{\sigma}$.

Now we have modeled the conditions that are satisfied by American options as an LCP, we can formulate the corresponding calibration problem as an MPCC. The goal of the calibration problem is to search for a volatility function that generates American options least deviated from the observed market prices. Therefore, a common choice of the objective function can be the Euclidean distance between the model generated theoretical option prices and the observed market prices. Denote the objective function as $\Phi(\bar{\sigma}, \mathbf{V})$, it can be mathematically expressed as

$$\Phi(\bar{\sigma}, \mathbf{V}) \equiv \sum_{l,m,n} \left(V_n^{m,l} - V_n^{\text{obs},m,l} \right)^2 . \tag{4.6}$$

Here, $V_n^{m,l}$ is the theoretical option price of an American option with strike price K^l, evaluated at time $m\delta t$ with spot price being $n\delta S$. $V_n^{\text{obs},m,l}$ is the corresponding option price observed in the market. The choice of the objective function $\Phi(\bar{\sigma}, \mathbf{V})$ is not unique. For example, if we want our estimated volatility parameter $\bar{\sigma}$ to be close to the historical volatility parameter $\bar{\sigma}^{\text{hist}}$, we can then add an extra penalty term $\lambda||\bar{\sigma} - \sigma^{\text{hist}}||^2$ to the objective function (4.6). In our experiments, we also use the following variant of equation (4.6) as our objective function:

$$\Phi(\bar{\sigma}, \mathbf{V}) \equiv \sum_{l,m,n} \left(\frac{V_n^{m,l} - V_n^{\text{obs},m,l}}{V_n^{\text{obs},m,l}} \right)^2 . \tag{4.7}$$

In general, due to the market consideration, we require the volatility function to remain in a reasonable range. We denote the feasible range of $\bar{\sigma}$ as Γ. One reasonable choice of Γ can be a hyper-rectangle. Given the objective function $\Phi(\bar{\sigma}, \mathbf{V})$ and Γ, our final formulation of the calibration problem becomes an MPCC of the following form:

$$\begin{array}{ll} \text{minimize} & \Phi(\bar{\sigma}, \mathbf{V}) \\ \text{subject to} & \bar{\sigma} \in \Gamma, \\ \text{and} & 0 \leq \mathbf{V} - \mathbf{\Psi} \perp \mathbf{d}(\bar{\sigma}) + \mathbf{M}_T(\bar{\sigma})\mathbf{V} \geq 0. \end{array} \tag{4.8}$$

The advantage of the above formulation is that it computes the volatility parameter $\bar{\sigma}$ and the option prices $\mathbf{V}$ jointly by taking into consideration the market information.

4.2.2 The MPCC of the Dupire System

In this work, we also consider the calibration problem of American options under the Dupire system (2.19). Similar to the formulation of the calibration problem under the local volatility model, we use the LCP (2.30) obtained by the finite element method as the complementarity constraints of the MPCC for the Dupire system. Recall that under the Dupire system, the American options satisfy the following sequence of LCPs. For $j = 1, 2, \ldots, R$,

$$0 \leq u^j \perp q^j + \mathbf{M}(\sigma)u^j \geq 0.$$

Here, $\mathbf{M}(\sigma) \equiv \mathbf{M} + dt\mathbf{A}(\sigma)$ and $q^j \equiv -(\mathbf{M} + dt\mathbf{B})u^{j-1} + dt\mathbf{F}$. Here, q^j does not depend on the parameter σ due to the discretization of the time variable t by the explicit finite difference method. It is worth mentioning that there are a few differences between the LCPs derived under the local volatility model and the Dupire system. Under the basic assumption of the Dupire system, the volatility parameter is a constant σ for American options with different strike prices and time to maturities. Therefore, the volatility parameter is a scalar instead of a vector in the Dupire system. Moreover, the other unknown vector u^j is no longer the option prices, but the time values of the corresponding American options. Therefore, for an MPCC formulated under the Dupire system (2.19), the unknown variables are the volatility constant σ and the time values of the American options u. Here, $u = (u^j)_{j=1}^R$ contains the time values of American options with different maturities and strike prices.

Similar to the formulation of the aggregated complementarity constraints under the local volatility model, we concatenate all the LCPs satisfied by the American options under the Dupire system together. The final formulation of the corresponding MPCC can be summarized as follows.

$$\begin{array}{ll} \text{minimize} & \Phi(\sigma, u) \\ \text{subject to} & \sigma \in \Gamma, \\ \text{and} & 0 \leq u \perp \mathbf{F}_T + \mathbf{M}_T(\sigma)u \geq 0. \end{array}$$

Here,

$$\mathbf{M}_T(\sigma) \equiv \begin{pmatrix} \mathbf{M}(\sigma) & & & & \\ -(\mathbf{M} + dt\mathbf{B}) & \mathbf{M}(\sigma) & & & \\ & -(\mathbf{M} + dt\mathbf{B}) & \ddots & & \\ & & \ddots & \mathbf{M}(\sigma) & \\ & & & -(\mathbf{M} + dt\mathbf{B}) & \mathbf{M}(\sigma) \end{pmatrix}$$

is the aggregated matrix of size $RQ \times RQ$, and

$$\mathbf{F}_T \equiv \begin{pmatrix} dt\mathbf{F} - (\mathbf{M} + dt\mathbf{B})u^0 \\ dt\mathbf{F} \\ \vdots \\ dt\mathbf{F} \end{pmatrix} \tag{4.9}$$

is the aggregated load vector of size RQ. Now that we have formulated the calibration problems of the volatility parameter as MPCCs, we devote the rest of the chapter to the development of numerical methods for solving MPCCs of this type.

4.3 Solution algorithms

As discussed in [LPR96] and [HP00], the complementarity constraints in an MPCC introduce both disjunction and non-smoothness to the solution $v(\sigma)$. The KKT optimality theory does not hold for an MPCC and the KKT multipliers in general do not exist. Due to the special structure of the problem, standard nonlinear programming theory does not work for this type of optimization problems. Comprehensive studies of this class of optimization problems have been made and several specialized methods have been developed. In [FJQ99], Facchinei, Jiang and Qi proposed a smoothing method to solve this type of problems. In [FLRS06], Fletcher, Leyffer, Ralph and Scholtes suggested using SQP methods to solve MPCCs. In the same paper, they also provided the local convergence results of the corresponding SQP methods. In [HP00], Huang and Pang presented two novel approaches for solving MPCCs related to the calibration problems of American options. In this thesis, we follow one of their approaches, an implicit programming algorithm (IMPA) to solve the MPCCs derived under both the Dupire system and the local volatility model. In addition, we propose a hybrid algorithm based on IMPA to further accelerate the convergence. The newly proposed algorithm allows us to solve an MPCC with large constraints within a reasonably short time.

4.3.1 An implicit programming algorithm

Before we discuss the details of IMPA, we begin with the review of some properties of the implicit function $v(\sigma)$. Although the properties are well explained in [CPS09, Chapter 7], it is useful to repeat them for a better understanding of the design of IMPA. Without loss of generosity, we consider a general MPCC with objective function $\Phi(\sigma, v)$ and the following constraints in the variables (σ, v):

$$0 \leq v \perp q(\sigma) + M(\sigma)v \geq 0, \text{ and } \sigma \in \Gamma. \tag{4.10}$$

Here, σ and v are vectors in $\mathbf{R}^I$ and $\mathbf{R}^J$, respectively . If the matrix $M(\sigma)$ in the complementarity constraints (4.10) is a P-matrix for any σ, then by applying the result of Theorem 2.2.4, we have that there is a unique solution $v(\sigma)$ to the complementarity system (4.10) for any given σ. Therefore, $v(\sigma) \in \mathbf{R}^J$ can be viewed as a function of $\sigma \in \mathbf{R}^I$.

The key idea of IMPA is to find a descent direction of the objective function. As such, studies of the local properties of $v(\sigma)$ when σ undergoes small perturbation are essential. The following local property describes the conditions that a directional derivative of $v(\sigma)$ satisfies and hence plays an important role in the design of IMPA. Denote

$$F(\sigma, v) \equiv q(\sigma) + M(\sigma)v, \tag{4.11}$$

and define the associated index sets $\alpha(\sigma)$, $\beta(\sigma)$ and $\gamma(\sigma)$ for any pair (σ, v).

$$\begin{aligned} \alpha(\sigma) &\equiv \{i : v_i(\sigma) > 0 = (q(\sigma) + M(\sigma)v)_i\}, \\ \beta(\sigma) &\equiv \{i : v_i(\sigma) = 0 = (q(\sigma) + M(\sigma)v)_i\}, \\ \gamma(\sigma) &\equiv \{i : v_i(\sigma) = 0 < (q(\sigma) + M(\sigma)v)_i\}. \end{aligned} \tag{4.12}$$

We summarize the local property of $v(\sigma)$ as follows.

Theorem 4.3.1. *Suppose that the associated matrix $M(\sigma) \in \mathbf{R}^{J\times J}$ in (4.10) is a P-matrix for any $\sigma \in \mathbf{R}^I$, then the function $v(\sigma)$ is Lipschitz continuous and directionally differentiable in its argument $\sigma \in \mathbf{R}^I$. The directional derivative of $v(\sigma)$ at any $\sigma \in \mathbf{R}^I$ along any direction $d\sigma \in \mathbf{R}^I$, denoted as $v'(\sigma; d\sigma)$, is the unique solution $dv \in \mathbf{R}^J$ to the following mixed LCP:*

$$\begin{aligned} &\left(J_\sigma F(\sigma, v(\sigma))d\sigma + M(\sigma)dv\right)_i = 0, \ \ \forall\, i \in \alpha(\sigma), \\ &0 \leq (dv)_i \perp (J_\sigma F(\sigma, v(\sigma))d\sigma + M(\sigma)dv)_i \geq 0, \ \ \forall\, i \in \beta(\sigma), \\ &(dv)_i = 0, \ \ \forall\, i \in \gamma(\sigma). \end{aligned} \tag{4.13}$$

Furthermore, it holds that

$$v(\sigma + d\sigma) = v(\sigma) + v'(\sigma; d\sigma) + o(||d\sigma||).$$

The proof and a detailed discussion of Theorem 4.3.1 can be found in [CPS09, Chapter 7]. According to Theorem 4.3.1, the directional derivative $v'(\sigma; d\sigma)$ is the solution to the mixed LCP (4.13) of a reduced size. Especially, when $\beta(\sigma)$ is empty, $v'(\sigma; d\sigma)$ is the solution of a system of linear equations. Hence, it is practical to use the following system of linear equations as an approximation to the mixed LCP (4.13):

$$\begin{aligned} &(J_\sigma F d\sigma + M(\sigma)dv)_i = 0, \ \ \forall\, i \in \alpha(\sigma) \cup \beta'(\sigma), \\ &(dv)_i = 0, \ \ \forall\, i \in \gamma(\sigma) \cup (\beta(\sigma)\backslash\beta'(\sigma)). \end{aligned} \tag{4.14}$$

Here, $\beta'(\sigma)$ can be any subset of $\beta(\sigma)$. With the above simplification of the constraints satisfied by the directional derivative, we transform the problem of searching for a descent direction of the objective function $\Phi(\sigma, v)$ into a convex quadratic program defined by

$$\begin{aligned} &\text{minimize} && (d\Phi_\sigma)^T d\sigma + (d\Phi_v)^T dv + \frac{1}{2} d\sigma^T W d\sigma \\ &\text{subject to} && \sigma + d\sigma \in \Gamma, \\ &\text{and} && (J_\sigma F d\sigma + M(\sigma)dv)_i = 0, \ \ \forall\, i \in \alpha \cup \beta', \\ & && (dv)_i = 0, \ \ \forall\, i \in \gamma \cup (\beta\backslash\beta'). \end{aligned} \tag{4.15}$$

Here, $d\Phi_\sigma \equiv \bigtriangledown_\sigma \phi(\sigma, v)$, $d\Phi_v \equiv \bigtriangledown_v \phi(\sigma, v)$, $J_\sigma F \equiv J_\sigma F(\sigma, v)$ and W is a given symmetric positive definite matrix of size $I \times I$.

Given the above generation of the search direction, we now can build IMPA as an iterative descent algorithm. At each iteration ν, we solve system (4.15) at σ^ν for a search direction $d\sigma$ which could lead to a potential decrease of the objective function. We then apply the Armijo inexact linear search rule (discussed in Section 3.3) to determine the step size. Also, a lower bound of 0.01 is set on the step size to prevent it from being too small. In our numerical experiments, we choose the matrix W of system (4.15) to be an identity matrix of size $I \times I$ for all iterations. Different choices of W at each iteration might affect the numerical performance of IMPA. However, the choice of an identity matrix for all iterations produces satisfactory numerical results in our experiments. A step-by-step summary of IMPA is presented in Algorithm 6.

To solve the calibration problems of American options under the local volatility model efficiently, we not only reduce the dimension of the formulated MPCC from $R(Q-1)$ to p_1p_2 by the special construction of the local volatility function, but also further modify the implementation of IMPA by decomposing the sub-problem (4.16) at each iteration into smaller systems. The decomposition greatly facilitates the computational capability of IMPA for solving MPCCs with large constraints. We use the MPCC derived under the local volatility model as an example to illustrate our modification of IMPA.

Notice that in MPCC (4.8), the volatility function is constructed by the unknown vector $\bar{\sigma}$. Hence,

```
Algorithm 6: IMPA
  Input: scalars $\rho \in (0,1)$ and $\gamma \in (0,1)$;
  initialize $\sigma^0 \in \Gamma$ and $\nu = 0$;
  set $W$ to be a symmetric positive definite matrix of order $I \times I$, with $I$ being the size of $\sigma^0$;
  repeat
      Step 1
      determine the index sets $\alpha^\nu$, $\beta^\nu$ and $\gamma^\nu$ associated with $(\sigma^\nu, v(\sigma^\nu))$;
      let $\beta'$ be an arbitrary subset of $\beta^\nu$;
      solve the convex quadratic program
          $\begin{aligned} &\text{minimize} && (d\Phi_\sigma^\nu)^T d\sigma + (d\Phi_v^\nu)^T dv + \frac{1}{2} d\sigma^T W d\sigma \\ &\text{subject to} && \sigma^\nu + d\sigma \in \Gamma, \\ &\text{and} && (J_\sigma F d\sigma + M(\sigma^\nu) dv)_i = 0, \ \ \forall\, i \in \alpha^\nu \cup \beta', \\ & && (dv)_i = 0, \ \ \forall\, i \in \gamma^\nu \cup (\beta^\nu \backslash \beta'); \end{aligned} \qquad (4.16)$
      record the search direction $d\sigma$ and the auxiliary vector $dv$;
      Step 2
      define the function $\sigma^\nu(\tau) \equiv \sigma^\nu + \tau d\sigma$ and let $v^\nu(\tau) \equiv v(\sigma^\nu(\tau))$;
      search for $l_\nu$, where $l_\nu$ is the smallest non-negative integer that satisfies either:
      (i) $\Phi(\sigma^\nu(\tau), v^\nu(\tau)) - \Phi^\nu \leq \gamma\tau \left((d\Phi_\sigma^\nu)^T d\sigma + (d\Phi_v^\nu)^T dv\right)$, where $\tau \equiv \rho^{l_\nu}$, or
      (ii) $\rho^{l_\nu} \leq 0.01$;
      set $\tau_\nu \equiv \max(0.01, \rho^{l_\nu})$;
      update $\sigma^{\nu+1} \equiv \sigma^\nu(\tau_\nu)$;
      $\nu \leftarrow \nu + 1$;
  until a stopping test is satisfied;
```

the unknown variable of this MPCC is $\bar{\sigma}$. According to Algorithm 6, the first step of IMPA involves the calculation of $J_{\bar{\sigma}} F$. From (2.24), we know that for the MPCC derived under the local volatility model,

$$\begin{aligned} F(\sigma, \mathbf{V}) = &\left((F_i^m(\sigma, \mathbf{V}))_{i=1}^{Q-1} \right)_{m=0}^{R-1} \\ =&\mathbf{d}(\sigma) + \mathbf{M}_T(\sigma)\mathbf{V}. \end{aligned}$$

Hence, for each entry $F_i^m(\sigma, \mathbf{V})$, we have

$$F_i^m(\sigma, \mathbf{V}) = d_i^m\ (\sigma) + \sum_{j=1}^{Q-1} \widehat{\mathbf{M}}_{ij}^m V_j^{m+1} + \sum_{j=1}^{Q-1} \mathbf{M}_{ij}^m V_j^m,$$

with

$$d_i^m = \begin{cases} \frac{1}{2}\left(\theta V_0^m + (1-\theta)V_0^{m+1}\right)\left(r - q - (\sigma_1^m)^2\right), & i = 1, \\ -\frac{1}{2}\left(\theta V_Q^m + (1-\theta)V_Q^{m+1}\right)\left((Q-1)(r-q) + (Q-1)^2(\sigma_{Q-1}^m)^2\right), & i = Q-1, \\ 0, & otherwise. \end{cases}$$

$$\widehat{\mathbf{M}}_{ij}^m = \begin{cases} -\frac{1}{2}(1-\theta)(\sigma_i^m)^2 i^2 - (1-\theta)\frac{(r-q)i}{2}, & i = j-1, \\ -\frac{1}{\delta t} + (1-\theta)(\sigma_i^m)^2 i^2, & i = j, \\ -\frac{1}{2}(1-\theta)(\sigma_i^m)^2 i^2 + (1-\theta)\frac{(r-q)i}{2}, & i = j+1 \\ 0, & otherwise. \end{cases}$$

$$\mathbf{M}_{ij}^m = \begin{cases} -\frac{1}{2}\theta(\sigma_i^m)^2 i^2 - \theta\frac{(r-q)i}{2}, & i = j-1, \\ \frac{1}{\delta t} + \theta(\sigma_i^m)^2 i^2 + r, & i = j, \\ -\frac{1}{2}\theta(\sigma_i^m)^2 i^2 + \theta\frac{(r-q)i}{2}, & i = j+1, \\ 0, & otherwise. \end{cases}$$

Therefore, $\frac{\partial F_i^m(\sigma,\mathbf{V})}{\partial \bar{\sigma}_{gk}}$ can be calculated explicitly as follows.

$$\begin{aligned} \frac{\partial F_i^m(\sigma,\mathbf{V})}{\partial \bar{\sigma}_{gk}} &= \frac{\partial d_i^m(\sigma)}{\partial \bar{\sigma}_{gk}} + \sum_{j=1}^{Q-1} \frac{\partial \widehat{\mathbf{M}}_{ij}^m}{\partial \bar{\sigma}_{gk}} V_j^{m+1} + \sum_{j=1}^{Q-1} \frac{\partial \mathbf{M}_{ij}^m}{\partial \bar{\sigma}_{gk}} V_j^m \\ &= \begin{cases} \frac{1}{2}\frac{\partial(\sigma_1^m)^2}{\partial\bar{\sigma}_{gk}}\left(-\theta V_0^m - (1-\theta)V_0^{m+1} + \theta(2V_1^m - V_2^m) + (1-\theta)(2V_1^{m+1} - V_2^{m+1})\right), & i = 1, \\ \frac{1}{2}(Q-1)^2\frac{\partial(\sigma_{Q-1}^m)^2}{\partial\bar{\sigma}_{gk}}\left(-\theta V_Q^m - (1-\theta)V_Q^{m+1} + \theta(2V_{Q-1}^m - V_{Q-2}^m) + (1-\theta)(2V_{Q-1}^{m+1} - V_{Q-2}^{m+1})\right), & i = Q-1, \\ \frac{1}{2}i^2\frac{\partial(\sigma_i^m)^2}{\partial\bar{\sigma}_{gk}}\left(\theta(2V_i^m - V_{i+1}^m - V_{i-1}^m) + (1-\theta)(2V_i^{m+1} - V_{i+1}^{m+1} - V_{i-1}^{m+1})\right), & otherwise. \end{cases} \end{aligned}$$

Recall that we also have

$$\begin{aligned} \sigma &= \left((\sigma_i^m)_{i=1}^{Q-1}\right)_{m=0}^{R-1} \\ &= \left(\left(\sum_{g,k} \bar{\sigma}_{gk}\phi_{gk}(i\delta S, m\delta t)\right)_{i=1}^{Q-1}\right)_{m=0}^{R-1}. \end{aligned}$$

Therefore, $\frac{\partial(\sigma_i^m)^2}{\partial\bar{\sigma}_{gk}}$ can be computed as

$$\frac{\partial(\sigma_i^m)^2}{\partial\bar{\sigma}_{gk}} = 2\sigma_i^m \phi_{gk}(i\delta S, m\delta t).$$

Now that we have calculated $J_{\bar{\sigma}}F$ explicitly, the first equality constraint of (4.16) in Algorithm 6 can be expressed as

$$\sum_{g,k} \frac{\partial F_i^m}{\partial \bar{\sigma}_{gk}} d\bar{\sigma}_{gk} + \left(\widehat{\mathbf{M}}_{ii-1}^m(\sigma) dV_{i-1}^{m+1} + \widehat{\mathbf{M}}_{ii}^m(\sigma) dV_i^{m+1} + \widehat{\mathbf{M}}_{ii+1}^m(\sigma) dV_{i+1}^{m+1} \right)$$
$$+ \left(\mathbf{M}_{ii-1}^m(\sigma) dV_{i-1}^m + \mathbf{M}_{ii}^m(\sigma) dV_i^m + \mathbf{M}_{ii+1}^m(\sigma) dV_{i+1}^m \right) = 0,$$

for all $mQ + i \in \alpha(\sigma)$.

Hence, by taking advantage of the expression of $J_{\bar{\sigma}}F$, we can decompose the constraints of (4.16) into constraints of smaller sizes in the implementation of IMPA. This modification greatly enhances IMPA's capability of solving MPCCs with constraints of large sizes. Specifically, we first decompose the the constraints of (4.16) into R smaller subgroups of constraints, each having $Q-1$ constraints. Starting from the last subgroup of $Q-1$ constraints, we calculate $\left(dV_i^{R-1}\right)_{i=1}^{Q-1}$ in terms of $d\bar{\sigma}$ by solving a system of linear equations. We then substitute $\left(dV_i^{R-1}\right)_{i=1}^{Q-1}$ by the expression of $d\bar{\sigma}$ in the second to last subgroup of $Q-1$ constraints. Again, we obtain the expression of $\left(dV_i^{R-2}\right)_{i=1}^{Q-1}$ as a function of $d\bar{\sigma}$ by solving a system of linear equations. We repeat the above procedure until all the entries of dV have been expressed as a function of $d\bar{\sigma}$. The total computational cost of this process is equivalent to solving R systems of linear equations of size $Q-1$. Notice that both the matrices $\mathbf{M}^m$ and $\widehat{\mathbf{M}}^m$ appeared in the constraints of the m-th subgroup are tridiagonal matrices. Hence, the update of $(dV_i^m)_{i=1}^{Q-1}$ can be performed within $O(Q)$ via the Gaussian elimination method. Once we write dV in terms of $d\bar{\sigma}$, we can rewrite the convex quadratic program (4.15) into a simply bounded optimization problem, and apply existing solvers to solve the corresponding optimization problem.

The same methodology can be applied to the MPCC derived under the Dupire system. Moreover, for an MPCC derived under the Dupire system, the unknown volatility parameter σ is a scalar. Therefore, applying the modified IMPA to the corresponding MPCC leads to solving a one-dimensional convex quadratic program, which can be solved by comparing the objective values at three (possibly two) points, the two end points of the feasible range and the point which solves $\bigtriangledown\Phi(\bar{\sigma}) = 0$ (given this point is in the feasible range). Given the above implementation of IMPA, we now are able to solve MPCCs with constraints of large sizes. Especially with the relaxed restriction on the size of the constraints, we now can solve MPCCs with finer discretizations and possibly obtain approximate solutions of the calibration problems with improved accuracy.

Other than IMPA, we also use two other approaches, a grid search method and a new hybrid method presented below to solve the formulated MPCCs in this work.

4.3.2 Grid search algorithm

Other than IMPA, one nature approach for solving the MPCCs derived from the calibration problems is to use a grid search method. As the name suggests, a grid search method is simply an exhaustive searching through a manually specified subset of the feasible range. In this thesis, we use a variant of the grid search method, a coordinate grid search method as the benchmark algorithm for solving the MPCCs derived from the calibration problems of American options. For simplicity, we call the coordinate grid search method as the grid search method in the rest of the work. To better illustrate our implementation of the grid search method, we apply it to the MPCC derived under the local volatility model and explain the method step by step.

Without loss of generosity, we assume that the feasible range of $\bar{\sigma}$ is a hyper-rectangle $\Gamma \in \mathbf{R}^{p_1 p_2}$. We divide the one-dimensional feasible interval of each coordinate i, $i = 1, 2, \ldots, p_1 p_2$, into I_i subintervals, and record the corresponding grid points as Σ_i. We then initialize our algorithm with a starting point $\bar{\sigma} \in \Gamma$ and search for an optimal solution by traversing through all coordinates. More precisely, we let the first coordinate of $\bar{\sigma}$ take values from Σ_1 while fixing the values of the rest coordinates. We evaluate the objective function at each point of Σ_1 and record the point that leads to the smallest objective value. This point is then used as the value of the first coordinate. We repeat the above procedure for each coordinate until we traverse through all the coordinates. Upon finishing one cycle of the grid search method, we update all coordinates of $\bar{\sigma}$. We repeat the cycle until a satisfactory result of the MPCC is achieved. The grid search method is summarized in Algorithm 7.

Algorithm 7: Coordinate grid search method

```
Input: a starting point $\bar{\sigma}^0 \in \mathbf{R}^{p_1 p_2}$, an integer parameter $\Delta_{\text{iter}} > 0$
and a discretization $\Sigma_1 \times \cdots \times \Sigma_{p_1 p_2}$ of $\Gamma \subseteq \mathbf{R}^{p_1 p_2}$;
set $\nu = 0$;
repeat
    set $i = 1$;
    repeat
        let the $i$-th coordinate of $\bar{\sigma}_i^\nu$ take values in $\Sigma_i$
        while fixing the values of other coordinates of $\bar{\sigma}^\nu$;
        evaluate the objective function at each point constructed above;
        update the $i$-th coordinate of $\bar{\sigma}_i^\nu$ by the point in $\Sigma_i$ that leads to the smallest objective value;
        $i \leftarrow i + 1$;
    until $i = p_1 p_2$;
    $\nu \leftarrow \nu + 1$;
until $\nu = \Delta_{\text{iter}}$;
```

The advantage of the grid search method is that it terminates within finite steps, which depend upon the discretization of Γ and the number of cycles. Also, the implementation of the grid search method is straightforward. However, the grid search method has its own limitations. The accuracy of the approximate solution generated by the grid search method depends heavily upon the discretization of Γ. The finer the grid is, the more accurate the solution will be. Therefore, the grid search method suffers from the curse of dimensionality. Especially for an optimization problem where the dimension of the parameter is large and the evaluation of the objective function is expensive, the grid search method in general does not produce satisfactory performance.

In our numerical experiments, we use a one-dimensional grid search method as a benchmark algorithm for solving MPCCs formulated under the Dupire system. To solve MPCCs of higher dimensions, for instance, the MPCCs derived under the local volatility model, we no longer use the grid search method in our experiments. Instead, we propose a hybrid method which combines the coordinate grid search method with IMPA for solving MPCCs of higher dimensions. We now present the hybrid method in the following section.

4.3.3 Hybrid algorithm

IMPA in general is efficient at searching for local solutions for MPCCs of high dimensions. However, it has it own limitations. Due to the simplification of the descent direction search (4.16) at each iteration of IMPA, there is no guarantee that the search direction $d\bar{\sigma}$ obtained is a descent direction of the objective function. Due to the challenge imposed by the non-convexity of the mixed LCP (4.13), there is no existing literature which addresses this issue. The numerical results of our experiments further suggest that IMPA is able to generate descent directions, which lead to substantial decrease of the objective function at its initial iterations. However, IMPA is not able to find such directions for its later iterations.

To remedy the slow convergence of IMPA, we propose a new hybrid algorithm, which employs both IMPA and a coordinate grid search method for solving MPCCs. The idea of this hybrid algorithm is motived by the fact that a grid search method helps IMPA to move away from the current iterate $\bar{\sigma}^{\nu}$ when it fails to find a descent direction or takes too small a step at the current iterate. By initializing the starting point of IMPA using the point generated by the grid search method, we are able to restart the searching of other regions for points with smaller objective values, and hence potentially accelerate the convergence.

We now present the hybrid algorithm in more detail. We start the algorithm by running a few iterations of IMPA. We stop iterations of IMPA when either the value of the objective function is smaller than a prescribed threshold value (this indicates that a satisfactory volatility function has been obtained), or there is no sufficient decrease of the objective function within these iterations. Notice that the choice of the number of IMPA iterations could presumably affect the performance of the hybrid algorithm. In our experiments, the number of IMPA iterations is set to be 30 and yields satisfactory results. We then run one cycle of the grid search method using the value of $\bar{\sigma}$ generated from the IMPA iterations as the starting point of the grid search. The cycle of IMPA iterations and a grid search step is repeated until a termination test is satisfied.

The grid search iterations of the above described hybrid method can be quite expensive when the dimension of $\bar{\sigma}$ is large. Therefore, to further improve the efficiency of our hybrid algorithm, instead of searching through all coordinates of $\bar{\sigma}$, we only search coordinates for which the objective function is sensitive; we call these influential coordinates. In our implementation, we adopt a heuristic strategy to determine the influential coordinates. To illustrate this heuristic, consider the calibration problem derived under the local volatility model. We first evaluate the option prices with a 0.2 uniform volatility function. Here, by a uniform volatility function with constant $\vartheta > 0$, we mean that all the elements of $\bar{\sigma}$ are fixed at ϑ. The evaluation of the option prices is equivalent to solving the LCP (4.4) with this definition of $\bar{\sigma}$. Next, for each coordinate of $\bar{\sigma}$, we increase its value by 1 percent and re-evaluate the option prices. If none of the option prices change by more than 1 basis point, then this coordinate is not considered to be an influential coordinate. After iterating through all the elements of $\bar{\sigma}$, we obtain the most influential coordinates Λ.

In our experiments we test the hybrid method on synthetic data of American options. The numerical results (see Table 4.12) suggest that the hybrid method does improve the convergence of IMPA, and is able to find $\bar{\sigma}$ resulting in a small object value within a reasonably short time. The fast convergence of the hybrid algorithm mainly arises from IMPA's capability of calculating effective descent directions during early iterations and the grid search method's ability to find better local minimizers (since it searches over a grid of the space). The hybrid method is formally stated as Algorithm 8.

4.4 Numerical experiments

In this section, we report the numerical performance of IMPA, the grid search method and the new hybrid algorithm for solving MPCCs derived under the Dupire system and the local volatility model. The experimental codes are written in MATLAB and C++. Specifically, IMPA is implemented in MATLAB and tested with MPCCs derived under both the Dupire system and the local volatility model. To better understand the performance of IMPA, the grid search method is tested with MPCCs derived under the Dupire system in MATLAB as a benchmark algorithm. As for the hybrid method, the IMPA iterations of the hybrid method are implemented in MATLAB while the grid search iterations of the hybrid method are implemented in C++. The hybrid method is tested for solving MPCCs derived under the local volatility model.

Before presenting the numerical results, we first provide the implementation details of each algorithm. For IMPA, the main computation involves the calculation of a search direction and the determination of a

Algorithm 8: Hybrid algorithm

```
Input: a starting point $\bar{\sigma}^0$, integer parameters $\Delta_{\text{impa,iter}} > 0$ and $\Delta_{\text{iter}} > 0$,
scalars $\epsilon^0 > 0$ and $\varsigma \in (0, 1)$ for the determination of the threshold value $\epsilon^\nu$,
the index set of the influential coordinates $\Lambda$,
and the corresponding discretization of the influential coordinates $\Pi_{i\in\Lambda}\Sigma_i$;
set $\nu = 0$;
repeat
    Step 1
    run $\Delta_{\text{impa,iter}}$ iterations of IMPA until either:
    (i) IMPA terminates, or
    (ii) the absolute difference of the objective values at two consecutive IMPA iterations is smaller
    than the threshold value $\epsilon^\nu > 0$;
    $\epsilon^\nu \leftarrow \epsilon^\nu \times \varsigma$;
    update $\bar{\sigma}^\nu$ with the result obtained from IMPA iterations;
    Step 2
    set $i = 1$;
    repeat
        let the $i$-th coordinate of $\bar{\sigma}^\nu_i$ take values in $\Sigma_i$
        while fixing the values of other coordinates of $\bar{\sigma}^\nu$;
        evaluate the objective function at each point constructed above;
        update the $i$-th coordinate of $\bar{\sigma}^\nu_i$ by the point in $\Sigma_i$ that leads to the smallest objective value;
        $i \leftarrow i + 1$;
    until $i =$ the number of elements in $\Lambda$;
    $\nu \leftarrow \nu + 1$;
until $\nu = \Delta_{\text{iter}}$;
```

step size at each iteration. In terms of the direction search step, we express dv in terms of $d\sigma$ and transform problem (4.15) into a strictly convex quadratic program in the variable $d\sigma$. For example, in the case of the local volatility model, once we calculate the j-th block of $\left(\left(dV_i^j\right)_{i=1}^{Q-1}\right)_{j=0}^{R-1}$, $\left(dV_i^j\right)_{i=1}^{Q-1}$ in terms of $d\bar{\sigma}$, we simply solve a system of linear equations and obtain the expression of the $(j-1)$-th block $\left(dV_i^{j-1}\right)_{i=1}^{Q-1}$ in terms of $d\bar{\sigma}$. When all the elements of $\left(\left(dV_i^j\right)_{i=1}^{Q-1}\right)_{j=0}^{R-1}$ have been calculated as a function of $d\bar{\sigma}$, we then use MATLAB's built-in quadratic programming solver (QUADPROG) to solve the corresponding convex quadratic program. In terms of the determination of the step size, evaluation of Φ at a sequence of points around (σ^ν, v^ν) is required at each iteration ν as presented in the second step of Algorithm 6. The evaluation involves solving an LCP for each point of the sequence. In the case that the associated matrix of the corresponding LCP is block tridiagonal, we again can decompose this LCP into a collection of LCPs of smaller sizes. By solving each smaller LCP separately via the two phase active-set method, we further improve the efficiency of IMPA. Similar to the implementation of the two phase active-set method discussed in Section 3.2, we set the stopping criterion as

$$\| \min(v, q + Mv) \|_\infty \leq \epsilon_{\text{sor}}$$

for solving $LCP(q,M)$. Here, ϵ_{sor} is the error tolerance level of the two phase active-set method. We set $\epsilon_{\text{sor}} = 10^{-6}$ in our implementation along with three other parameters $\omega = 1.5$, $k_{\text{sor}} = 3$ and $\Delta_{\text{ac}} = 20$ throughout our experiments. The termination test for IMPA is given as

$$\min\left(\| \, d\sigma \, \|_\infty, \Phi(\sigma)\right) \leq \epsilon_{\text{impa}}. \tag{4.17}$$

IMPA is terminated when either the search direction $d\sigma$ is smaller than a prescribed threshold value or a small objective value has been achieved. Also, a maximum number of 200 iterations is also imposed in our implementation. If after 200 iterations, the current iterate σ still does not satisfy the termination condition (4.17), we use σ^ν, which results in the smallest objective so far as the final output of IMPA.

For the hybrid method, let us consider the case of solving an MPCC derived under the local volatility model. In terms of the implementation of the IMPA iterations in the hybrid algorithm, it remains the same as that of the modified IMPA discussed in the previous paragraph. In terms of the implementation of the grid search step in the hybrid algorithm, for a give discretization of the feasible range Γ: $\Sigma_1 \times \cdots \Sigma_{p_1p_2}$, we need to evaluate a total number of $|\Sigma_1| \times \cdots |\Sigma_{p_1p_2}|$ LCPs at each iteration. We again employ the two phase active-set method to solve all LCPs per iteration. Notice that the choice of the influential coordinates Λ could presumably affect the practical performance of the hybrid algorithm. As discussed in Section 4.3.3, we determine the influential coordinates by using a 0.2 uniform volatility function in our problems. This heuristic method and the generated influential coordinates yield fairly satisfactory results in our numerical experiments. Now we have presented the numerical techniques involved in the implementation of the three algorithms, we report the numerical performance of the algorithms in the section below.

4.4.1 Calibration problems under the Dupire system

In this part, we consider the calibration problems of American put options under the Dupire system (2.19). Specifically, we consider calibrating the volatility parameter of a 0.5 years ($T = 0.5$) at the money (ATM) American put option with a spot price $S = 100$. Here, by at the money, we mean that the option's strike price is identical to the spot price of the underlying asset. Following the assumption of the Dupire system, we assume the risk-free interest rate of the market and the dividend of the underlying asset are constants

$$r = 0.05, \; q = 0.$$

We only consider a special class of the Dupire system, in which the jump density $p(z) = 0$ in this experiment. In this case, the jump operator $\mathcal{B} = 0$ in complementarity system (2.19) and the jump matrix $\mathbf{B}$ of LCP

(2.30) is a zero matrix. We set the computational domain of the log of the strike price x as

$$\underline{x} = -0.3, \ \bar{x} = 0.3.$$

For the practical purpose, it is reasonable to set an upper bound of 0.6 and a lower bound of 0.1 for the volatility parameter σ. In addition, we set the error tolerance level of IMPA as 10^{-6} together with two other parameters used in the determination of the step size as

$$\rho = 0.5, \ \gamma = 0.0005.$$

Since we are calibrating the volatility parameter of the ATM American put option, the objective function takes the following simple form:

$$\Phi(\sigma) = (u_{\text{ATM}} - u_{\text{ATM}}^{\text{obs}})^2. \tag{4.18}$$

Here, the observed price $u_{\text{ATM}}^{\text{obs}}$ is the price of the ATM put option calculated by solving LCP (2.30) at a prescribed volatility value. According to the P-property of the matrix $\mathbf{M} + dt\mathbf{A}(\sigma)$, there is a unique $u(\sigma)$ which is the solution of the corresponding LCP (2.30) at a given σ. Therefore, we would expect both IMPA and the grid search method to reproduce the prescribed volatility. Notice that under the Dupire system, the convex quadratic program obtained from system (4.13) is of dimension one. Therefore, the coordinate grid search method is simply a one-dimensional grid search method. For a fair comparison of IMPA and the grid search method, we discretize the feasible range Γ of σ into $(n_{\text{LCP}} - 1)$ evenly spaced subintervals, where n_{LCP} is the total number of LCPs solved by IMPA. We then evaluate the objective function at all end points of the subintervals in the grid search method. The end point with the smallest objective value is deemed as optimal for the grid search method.

Both IMPA and the grid search method are tested for solving the MPCCs derived under the Dupire system with different initial points and discretizations of time and space. For each method, we report its computational time in seconds (CPU) and the absolute differences between the recovered volatility values and the prescribed volatility values.

Case 1: space discretization $Q = 30$, time discretization $R = 10$

In this experiment, the prescribed volatility parameter is set to be 0.2. We calculate the corresponding price of the ATM put option by solving LCP (2.30) at $\sigma = 0.2$ with $\mathbf{B} = 0$. This value is computed as 4.6458 and used in the objective function (4.18). As discussed in the previous paragraph, we let both IMPA and the grid search method solve same number of LCPs in the computation of the MPCCs

	IMPA			Grid		
Initial point	Cal σ	Err	CPU(s)	Cal σ	Err	CPU(s)
0.1	0.2000	0.0000	0.3966	0.2008	0.0008	0.4938
0.5	0.2000	0.0000	1.5157	0.2000	0.0000	1.2761

Table 4.1: The Dupire system: Q=30, R=10.

	IMPA			Grid		
Initial point	Cal σ	Err	CPU(s)	Cal σ	Err	CPU(s)
0.1	0.2000	0.0000	9.2083	0.2004	0.0004	13.1768
0.5	0.2000	0.0000	47.2579	0.2000	0.0000	40.2076

Table 4.2: The Dupire system: Q=100, R=40.

derived under the Dupire system. According to our numerical results, when the initial points are 0.1 and 0.5, the total numbers of LCPs solved by IMPA are 119 and 320, respectively. Therefore, we divide the feasible range of σ into 118 and 319 evenly spaced subintervals for 0.1 and 0.5 initial points in the implementation of the grid search method. Ideally, we would like both methods to recover the prescribed $\sigma = 0.2$. Hence, we use the absolute difference between the calculated σ and the prescribed σ to measure the accuracy of the algorithms. More detailed results are presented in Table 4.1.

Case 2: space discretization $Q = 100$, time discretization $R = 40$

In this experiment, we test both IMPA and the grid search method with a finer discretization of time and space. Similar to the previous test, the prescribed volatility parameter is set to be 0.2. We again calculate the corresponding price of the ATM put option by solving LCP (2.30) at $\sigma = 0.2$ with $\mathbf{B} = 0$. This value is computed as 4.6539 and used in the objective function (4.18). We again record the total numbers of LCPs computed by IMPA with different initial points given the finer discretization of space and time. Our numerical results suggest that when the initial points are 0.1 and 0.5, the total numbers of LCPs solved by IMPA are 1077 and 3254, respectively. Therefore, we divide the feasible range of σ into 1076 and 3253 evenly spaced subintervals for 0.1 and 0.5 initial points, respectively in the implementation of the grid search method. The increased total numbers of LCPs are in accordance with our expectation since we use a finer discretization of space and time in this test. Similar to Table 4.1, we use the absolute difference between the calculated σ and the prescribed σ as the measurement of the accuracy for both algorithms. Detailed numerical performance of the two algorithms is presented in Table 4.2.

As suggested by Table 4.1 and Table 4.2, the performance of IMPA and the grid search method varies along with the space and time discretizations, as well as the initial points used in the initialization of

IMPA. According to our tests, IMPA outperforms the grid search method both in terms of accuracy and computational time for the two different discretizations when the initial point of IMPA is 0.1. Both IMPA and the grid search method are able to recover the prescribed σ when the initial point of IMPA is 0.5. However, the grid search method is able to reproduce the prescribed volatility within shorter times for the two discretizations. One possible explanation for the difference in two cases could be that when σ is small, the corresponding LCPs can be solved efficiently by the two phase active-set method. However, it is more costly to solve the LCPs when σ is large. Therefore, IMPA outperforms the grid search method when the initial point is close to the prescribed σ, and IMPA needs to solve less LCPs with large σ than the grid search method. The grid search method outperforms IMPA when the initial point is far from the prescribed σ, and IMPA needs to solve more LCPs with large σ than the grid search method. Overall speaking, both IMPA and the grid search method work equally well for solving MPCCs derived under the Dupire system.

4.4.2 Calibration problems under the local volatility model

In this section, we consider the calibration problems of American put options under the local volatility model. Due to the high dimensionality of the problems, we mainly use IMPA to solve MPCCs of this class. IMPA is tested with both synthetic data and market data: American index put options on the Standard & Poor's 100 index. In this section, we also test the new hybrid algorithm with the same set of synthetic data as IMPA. We now present the details of the data and the numerical performance of the two methods in the following sections.

Numerical performance of IMPA

In this part, we study the numerical performance of IMPA for solving MPCCs derived under the local volatility model. We test IMPA with both synthetic data generated by solving LCP (4.4) with a given uniform volatility function and market data of the American put options on the Standard & Poor's 100 index. In addition, we further split the market data into two subsets, training data and testing data. We calibrate the volatility parameter by applying IMPA to the training data, and use the calculated volatility parameter to compute the theoretical option prices for the testing data. The distance between the theoretical prices and the market prices of the testing data is then used to measure the effectiveness of IMPA.

Case 1: calibration problems with synthetic data

In this experiment, we consider the calibration problem of 14 American put options with strike prices $K = 420, 440, \ldots, 680$. The maturities of all 14 options are the same and given as 0.3 years ($T = 76/252$). We assume the current spot price is 590, and the risk-free interest rate of the market and the

K	420	440	460	480	500	520	540
Price	0.0183	0.0779	0.2698	0.7812	1.9343	4.1824	8.0446
K	560	580	600	620	640	660	680
price	21.2157	30.2530	41.7355	54.4589	68.7711	84.4186	96.0582

Table 4.3: Local volatility model: option prices with 0.2 uniform volatility function.

dividend of the underlying asset are constants

$$r = 0.0031,\ q = 0.0214.$$

For the practical purpose, we restrict the stock variable S on a finite computational domain with

$$S_{\min} = 0,\ S_{\max} = 800.$$

We set an upper bound of 0.6 and a lower bound of 0.1 for $\bar{\sigma}$, and let the space discretization be $Q = 800$ and the time discretization be $R = 50$ in this experiment. For each option with a fixed strike price K, we calculate the observed price $V^{\text{obs},K}$ by solving LCP (4.4) with a 0.2 uniform function. The prices of the 14 different options are summarized in Table 4.3.

As discussed in Section 4.1, we use the values $V^{\text{obs},K}$ to define the least-square objective function. Other than using the sum of the squares of the difference between the calculated theoretical price $V^{\text{cal},K}$ and the observed price $V^{\text{obs},K}$ of each option, we use the sum of the squares of the relative error $\left(V^{\text{obs},K} - V^{\text{cal},K}\right)/V^{\text{obs},K}$ of each option as the objective function in this experiment. Mathematically, the objective function takes form

$$\Phi(\bar{\sigma}) = \sum_K \left((V^{\text{obs},K} - V^{\text{cal},K})/V^{\text{obs},K}\right)^2. \tag{4.19}$$

The advantage of using such an objective function is that after the normalization of the errors, each option now has equal influence on the objective value. Hence, the volatility function obtained by IMPA is less likely to favor options with high prices. As for the construction of the volatility function, we use a 15×3 evenly spaced mesh to fully represent the 800×50 different points in the volatility function. Here, the space discretization is $p_1 = 15$ and the time discretization is $p_2 = 3$. Therefore, the dimension of the calibration problem is 45.

In terms of the implementation of IMPA, the error tolerance level of IMPA is set as 10^{-6} along with

Initial point 1	0.18× (1,1,. . .,1)
Initial point 2	0.22 × (1,1,. . .,1)
Initial point 3	0.40× (1,1,. . .,1)
Initial point 4	(0.60 : −0.03 : 0.18) × (0.60 : −0.03 : 0.18) × (0.60 : −0.03 : 0.18)

Table 4.4: Local volatility model: initial points of synthetic data.

K	Initial point 1	Initial point 2	Initial point 3	Initial point 4
420	-0.0644	-0.0121	-0.1015	-0.1049
440	0.0099	0.0140	-0.0026	0.0050
460	0.0337	0.0257	0.0337	0.0370
480	0.0300	0.0275	0.0403	0.0330
500	0.0234	0.0249	0.0330	0.0243
520	0.0177	0.0201	0.0254	0.0179
540	0.0134	0.0149	0.0194	0.0139
560	0.0099	0.0104	0.0131	0.0108
580	0.0070	0.0067	0.0080	0.0079
600	0.0045	0.0040	0.0043	0.0050
620	0.0023	0.0020	0.0017	0.0025
640	0.0005	0.0007	0.0006	0.0006
660	-0.0006	0.0000	0.0000	-0.0007
680	-0.0011	-0.0002	-0.0009	-0.0017
$\Phi(\bar{\sigma})$	0.0078	0.0032	0.0154	0.0148

Table 4.5: Local volatility model: relative errors of synthetic data.

two other parameters

$$\rho = 0.2,\ \gamma = 0.0005$$

used in the determination of the step size. In our experiment, we apply IMPA to the corresponding MPCCs with four different initial points $\bar{\sigma}^0$ (summarized in Table 4.4). For each initial point, we report the relative error of each option and the value of the objective function at the termination of IMPA. Details of the numerical results are presented in Table 4.5

It is interesting to note that in all four tests, IMPA is able to reproduce observed option prices closely when the strike prices are large than the spot price. Specifically, the relative errors are less than 0.5 percent for options with strike prices greater than 590 in all examples. Moreover, the resulting objective values are quite small when the initial points are close to the constant $\bar{\sigma}$ used to generate the observed market prices. Overall, IMPA is able to recover the observed prices quite well given different initial points.

Case 2: calibration problems with full market data

In this experiment, we test IMPA on the calibration problem of 14 American index put options on the

K	420	440	460	480	500	520	540
Price	1.30	1.75	1.80	2.70	2.95	5.29	7.40
K	560	580	600	620	640	660	680
price	10.90	17.00	26.00	39.80	56.10	74.50	94.10

Table 4.6: Local volatility model: option prices of S&P100.

Standard & Poor's 100 index. The strike prices of the options are $K = 420,\ 440, \ldots,\ 680$, and all the options share the same time to maturity $T = 76/252$. The current S&P100 index is at 590, and the risk-free interest rate of the market and the dividend of the S&P100 are

$$r = 0.0031,\ q = 0.0214.$$

We list the market prices of 14 put options in Table 4.6. All the data are quoted directly from Bloomberg's terminal.

We again apply IMPA to the MPCC (4.8) with the market data listed above. Similar to the previous example, we restrict the stock variable S on a finite computational domain with

$$S_{\min} = 0,\ S_{\max} = 800.$$

We again set an upper bound of 0.6 and a lower bound of 0.1 for $\bar{\sigma}$, and let the space discretization be $Q = 800$ and the time discretization be $R = 50$ in this experiment. The objective function take the same form as (4.19) with $V^{\text{obs},K}$ being the observed market prices of the American put options listed in Table 4.6. As for the construction of the volatility function, we again use a 15×3 evenly spaced mesh to fully represent 800×50 different grid points in the volatility function. Here, the space discretization is $p_1 = 15$ and the time discretization is $p_2 = 3$.

In terms of the implementation of IMPA, the error tolerance level of IMPA is set as 10^{-6} along with two other parameters

$$\rho = 0.2,\ \gamma = 0.0005$$

used in the determination of the step size. We apply IMPA to the corresponding MPCCs with two different initial points $\bar{\sigma}^0$, which are summarized in Table 4.7. For each initial point, we report the relative error of each option and the value of the objective function at the termination of IMPA. Details of the numerical performance of IMPA with the market data are presented in Table 4.8.

According to Table 4.8, IMPA presents similar performance compared to that of the previous exper-

Initial point 1	0.2× (1,1,. . .,1)
Initial point 2	(0.6:-0.03:0.18)×(0.57:-0.03:0.15)×(-0.54:-0.03:0.12)

Table 4.7: Local volatility model: initial points of market data.

K	Initial point 1	Initial point 2
420	-0.2377	-0.2014
440	-0.1766	-0.1560
460	0.1093	0.1251
480	-0.0064	0.0030
500	0.2152	0.2251
520	-0.0490	-0.0414
540	-0.0213	-0.0147
560	0.0115	0.0154
580	0.0133	0.0166
600	0.0287	0.0331
620	0.0008	0.0039
640	-0.0003	0.0001
660	-0.0017	-0.0026
680	-0.0049	-0.0058
$\Phi(\bar{\sigma})$	0.1500	0.1349

Table 4.8: Local volatility model: relative errors of market data.

iment. It is able to reproduce theoretical option prices which are considerably close to the observed market prices. Especially for options with strike prices greater than the current index, the relative errors are less than 5 percent for the two different initial points.

Case 3: calibration problems with training market data

Numerical results have demonstrated IMPA's capability of generating option prices that are close to their observed prices with both synthetic data and market data. However, we are also interested in the predictive ability of the volatility function generated by IMPA. In the following experiment, we divide the aforementioned market data (Table 4.6) into two groups, training data and testing data. We use IMPA to solve the MPCC derived under the local volatility model with the training data. The obtained volatility function is then used to compute the theoretical prices of the put options in the testing data. Ideally, we would like IMPA to generate a volatility function which yields theoretical prices that are close to the market prices of the options in the testing data. Therefore, we report both the absolute difference and the relative difference between the market price and the theoretical price of each option in the testing data as measurements of IMPA's performance.

In our example, the data are divided as follows.

- training data: options with strike prices $K = 420, 440, 460, 500, 540, 560, 600, 640, 660, 680$.

K	Rel err
420	0.2000
440	0.1625
460	0.0966
500	0.1411
540	0.0548
560	0.0057
600	0.0257
640	0.0021
660	0.0012
680	0.0040
$\Phi(\bar{\sigma})$	0.0993

Table 4.9: Local volatility model: relative errors of training data.

K	Mkt price	Cal price	Abs diff	Rel err
480	2.70	2.6742	0.0258	0.0096
520	5.20	4.7697	0.4303	0.0827
580	17.00	17.4890	0.4890	0.0287
620	39.80	39.7381	0.0619	0.0015

Table 4.10: Local volatility model: relative errors of testing data.

- testing data: options with strike prices $K = 480, 520, 580, 620$.

The parameters used in the implementation of IMPA and the construction of the volatility function remain the same as those used in the previous two experiments (Case 1 and Case 2 of Section 4.4.2). In this experiment, we set the initial point $\bar{\sigma}^0$ to be the 0.2 uniform function. For the training data, the relative error of each option is reported in Table 4.9. For the testing data, both the absolute difference between the theoretical price and the market price, and the relative error of each option are reported in Table 4.10.

According to the numerical results, we note that IMPA is able to recover the market prices relatively well for options with large strike prices for both the training and the testing set. The experiment has demonstrated IMPA's predictive capability and further suggests its possible use as an effective tool for pricing and hedging of index options in practice.

Numerical performance of the hybrid method

In this part, we study the numerical performance of the newly proposed hybrid algorithm for solving the MPCCs derived under the local volatility model. We conduct the test with the same set of synthetic put option data (Table 4.3). In terms of the construction of the volatility function, we again use a 15×3

Initial point 1	0.10× (1,1,...,1)
Initial point 2	0.40 × (1,1,...,1)

Table 4.11: Local volatility model: initial points of synthetic data for the hybrid algorithm.

Loop	Step	Initial point 1	Initial point2	CPU(min)
Loop 1	IMPA	0.7579	0.7001	27
	Grid	0.0113	0.0113	54
Loop 2	IMPA	0.0058	0.0057	81
	Grid	0.0035	0.0035	54

Table 4.12: Local volatility model: computational time for the hybrid algorithm

evenly spaced mesh. In terms of the implementation of the IMAP iterations, the error tolerance level and the parameters used in the determination of the step size remain the same as those used in Case 1 and Case 2. We impose a maximum number of iterations $\Delta_{\text{impa,iter}} = 30$ on the IMPA iterations at each loop ν. Additionally, if the absolute difference between the two corresponding objective values is smaller than a prescribed threshold value $\epsilon^{\nu} > 0$ for two consecutive IMPA iterations, then the IMPA iterations is terminated immediately. In our example, we set

$$\epsilon^0 = 0.1, \; \varsigma = 0.01.$$

Once IMPA is terminated, we start the grid search method with $\bar{\sigma}$ generated by the IMPA iterations. In terms of the implementation of the grid search method, we use the heuristic method discussed in Section 4.3 to determine the influential coordinates. The influential coordinates in our example are calculated as

$$(6 : 1 : 12, \; 22 : 1 : 27, \; 36 : 1 : 42).$$

It is interesting to notice that the influential coordinates correspond to the points having $m\delta S$ close to the spot price 590. For each influential coordinate, we discretize the feasible range $[0.1, 0.6]$ into 20 evenly spaced subintervals and evaluate the objective function at each end point of the 20 subintervals. We test the hybrid algorithm with two different initial points listed in Table 4.11 for the evaluation of its performance. Note that in our implementation of the hybrid algorithm, IMPA iterations are implemented in MATLAB, and the grid search iterations are implemented in C++. We report both the objective value $\Phi(\bar{\sigma})$ and the corresponding computational time (CPU) of the two steps at each loop in Table 4.12. We also present the relative error of each option at the termination of the hybrid algorithm in Table 4.13.

The results of Table 4.12 and Table 4.13 suggest that the new hybrid algorithm is quite efficient in finding

K	Initial point 1	Initial point 2
420	-0.0388	-0.0388
440	-0.0319	-0.0319
460	0.0066	0.0066
480	0.0193	0.0193
500	0.0178	0.0178
520	0.0076	0.0076
540	-0.0024	-0.0024
560	-0.0075	-0.0075
580	-0.0082	-0.0082
600	-0.0056	-0.0056
620	-0.0017	-0.0017
640	0.0002	0.0002
660	0.0010	0.0010
680	0.0012	0.0012
$\Phi(\bar{\sigma})$	0.0035	0.0035

Table 4.13: Local volatility model: relative errors of synthetic data for the hybrid algorithm.

$\bar{\sigma}$ that yields satisfactory objective value. Given the two initial points in Table 4.12, it outperforms IMPA by finding $\bar{\sigma}$ which generates smaller objective values. Moreover, the hybrid algorithm is able to recover the observed price more closely than IMPA for the put option with small strike prices $K = 420, 440$, etc. Overall, the hybrid algorithm is an efficient algorithm for solving MPCCs under the local volatility model.

Chapter 5

Doubly uni-parametric MPCC

5.1 Introduction

For this part of our study, we consider the class of doubly uni-parametric MPCCs given by

$$\begin{aligned}&\text{minimize}_{\sigma,z}\ f(\sigma,z)\\&\text{subject to } \sigma\in\Gamma\equiv[\underline{\sigma},\bar{\sigma}],\\&\text{and } 0\leq z\perp q+\sigma p+(M+\sigma N)z\geq 0,\end{aligned}\tag{5.1}$$

where $f:\mathbf{R}^{n+1}\to\mathbf{R}$ is a differentiable scalar-valued objective function, $\underline{\sigma}<\bar{\sigma}$ are positive lower and upper bounds on the scalar variable σ, q and p are given vectors in $\mathbf{R}^n$, and M and N are positive definite matrices of order n that are not necessarily symmetric. The doubly uni-parametric complementarity constraints in (5.1) originate from the pricing of American options under the BSM model. Specifically, if we discretize the complementarity system (2.10) under the assumption that the volatility is constant, we obtain the following LCP:

$$0\leq z\perp q+\zeta^2 p+(M+\zeta^2 N)z\geq 0.\tag{5.2}$$

Here, ζ is the constant volatility of the underlying asset, and M and N are tridiagonal positive definite matrices. It is now easy to see that these are the doubly uni-parametric complementarity constraints of the form given in (5.1) with $\sigma=\zeta^2$.

As mentioned earlier, a comprehensive study of MPCCs along with computational techniques for solving them can be found in [LPR96]. However, instead of solving the optimization problem (5.1) as a general MPCC, we develop a specialized algorithm that is designed to take advantage of the structure of the doubly uni-parametric complementarity constraints. Notice that the complementarity condition in (5.1) given by

$$z^T\left(q+\sigma p+(M+\sigma N)z\right)=0$$

is a cubic polynomial equation in the pair (σ, z), and the inequality constraint

$$q + \sigma p + (M + \sigma N)z \geq 0 \tag{5.3}$$

is non-convex due to the bilinear term σz. These properties of our problem formulation necessitate the development of new theories and methodologies. Specifically, the key to our algorithm is to take advantage of the structure in the LCP constraints in (5.1) by linearizing the product σz.

The remainder of this chapter is organized as follows. In Section 5.2 we give a review of existing algorithms for solving a general MPCC (5.1) that include IMPA and a one-dimensional grid search method. In Section 5.3 we then consider a new algorithm for solving a doubly uni-parametric MPCC in the case that the associated matrices are positive definite. To extend our method to a more general class of problems, we consider in Section 5.4 a modification of the algorithm presented in Section 5.3 that is applicable when the matrices are merely positive semi-definite.

5.2 Existing algorithms

We present two existing algorithms for solving the doubly uni-parametric MPCC: IMPA and one-dimensional grid search method. The main reason for introducing these algorithms is for numerical comparisons with the new algorithms that we present in the remainder of this chapter.

The IMPA, which is a gradient-based algorithm, was introduced in Section 4.3.1 in the context of solving MPCCs. Since doubly uni-parametric MPCCs are contained in the class of MPCC, IMPA may be directly applied (Algorithm 6). We mention, however, that IMPA lacks a theoretical justification, unlike the methods that we present in Sections 5.3 and 5.4.

The second method that we consider is a brute force strategy based on a one-dimensional grid search. Although heuristic in nature, it has the advantage that it is very simple to describe and implement. To motivate the method, let us assume that the matrices M and N in (5.1) are positive definite. It then follows from [CPS09, Theorem 3.1.6] that the complementarity constraints in (5.1) have a unique solution $z(\sigma)$ for a given σ (In fact, we will show in Lemma 5.3.6 that $z(\sigma)$ is a piece-wise smooth function). Therefore, the optimization problem (5.1) is equivalent to the univariate minimization problem

$$\text{minimize}_{\sigma \in \Gamma}\ g(\sigma) \equiv f(\sigma, z(\sigma))$$

with the implicit function $g(\sigma)$ being highly non-convex. The one-dimensional grid search now proceeds by dividing the feasible range of σ into m small sub-intervals and evaluating the objective function at the corresponding $m+1$ grid points. The point with the smallest objective function is then considered as an approximate solution to the original problem (5.1). The computational time of the grid search method depends linearly on the size of the grid and the time required to compute $z(\sigma)$, i.e., solving $LCP(q+\sigma p, M+\sigma N)$. Details of the grid search algorithm are presented in Algorithm 9.

Algorithm 9: One-dimensional grid search method

```
**Input**: two matrices $M \in \mathbf{R}^{n\times n}$ and $N \in \mathbf{R}^{n\times n}$, two vectors $q \in \mathbf{R}^n$ and $p \in \mathbf{R}^n$, an integer $m > 0$;
**Step 1** divide the feasible range $\Gamma$ into $m$ evenly spaced sub-intervals with $m+1$ grid points $\{\sigma_i\}_{i=0}^m$;
**Step 2** initialize $i = 0$;
**repeat**
    compute $z^i \in SOL(q + \sigma_i p, M + \sigma_i N)$;
    evaluate $f(\sigma_i, z^i)$ with $f$ defined in (5.1);
    $i \leftarrow i + 1$;
**until** $i > m$;
**Step 3** choose $(\sigma_*, z^*) \in \{(\sigma_i, z^i)\}_{i=0}^m$ such that $f(\sigma_*, z^*) \le f(\sigma_i, z^i)$ for all $i = 0, 1, \ldots, m$,
and use $(\sigma_*, z^*)$ as the approximate solution of the grid search method
```

A nice feature of the one-dimensional grid search given by Algorithm 9 is that existing LCP methods may be used to compute z^i for the given grid point σ_i. On the negative side, however, as the grid becomes finer the required computation may become quite substantial.

5.3 A new algorithm for the positive definite case

In this section we describe our new method for the positive definite case and prove convergence of the resulting algorithm. The convergence analysis that we employ uses the fact that the problem matrices M and N are positive definite. Nonetheless, this serves as an important class of problems as discussed in Section 5.1 and includes American options pricing problems under the BSM model.

5.3.1 Formal statement of the algorithm

In this part we propose an innovative approach to solve the doubly uni-parametric MPCC (5.1) in the case that the associated matrices are positive definite by exploiting the special structure of the complementarity constraints

$$0 \le z \perp q + \sigma p + (M + \sigma N)z \ge 0. \tag{5.4}$$

The algorithm searches for directions which could potentially lead to the reduction of the objective function by solving a sequence of sub-problems of a special form. We present the construction of the sub-problems as follows. We start with an initial guess $\sigma_0 \in \Gamma$. For this given σ_0, we solve for z^0, which is the solution to the $LCP(q + \sigma_0 p, M + \sigma_0 p)$. Next, we use the pair (σ_0, z^0) to form sub-problem

$$\begin{aligned} &\text{minimize}_{\sigma,z}\ f(\sigma, z) + \frac{c}{2}[(\sigma - \sigma_0)^2 + \|z - z^0\|^2] \\ &\text{subject to } \sigma \in [\underline{\sigma}, \bar{\sigma}], \\ &\text{and } 0 \le z \perp (q - \sigma_0 N z^0) + \sigma(p + N z^0) + (M + \sigma_0 N)z \ge 0. \end{aligned} \tag{5.5}$$

Since,

$$\begin{aligned} &(q - \sigma_0 N z^0) + \sigma(p + N z^0) + (M + \sigma_0 N)z \\ &= q + \sigma p + (M + \sigma N)z - (\sigma - \sigma_0)N(z - z^0). \end{aligned}$$

it is easy to see that the complementarity constraints in (5.5) are close approximations to the doubly uniparametric complementarity constraints in (5.4) for (σ, z) around (σ_0, z^0). Once we solve the sub-problem (5.5), we record its solution as $(\sigma_1, z^{0.5})$. We then use σ_1 to solve for z^1, which is the solution to the $LCP(q + \sigma_1 p, M + \sigma_1 N)$. The pair (σ_1, z^1) is used to construct the next sub-problem by replacing the pair (σ_0, z^0) in (5.5). By repeating the above procedure, we therefore construct a sequence of sub-problems. At the same time, we also find a sequence of search directions, $d\sigma_i = \sigma_{i+1} - \sigma_i$, which could potentially leads to the decrease of the objective function and an approximate solution of the original problem (5.1). We only stop the computation of the sub-problems and the search of directions when the resulting pair (σ, z) satisfies certain stopping criterion and is deemed as satisfactory. By using the above construction of the sub-problems, we can now present our new algorithm for the positive definite case in Algorithm 10.

Algorithm 10: Linearized approach for an MPCC with positive definite matrices

```
**Input**: positive definite matrices $M \in \mathbf{R}^{n\times n}$ and $N \in \mathbf{R}^{n\times n}$,
vectors $q \in \mathbf{R}^n$ and $p \in \mathbf{R}^n$, and a scalar $c > 0$;
set $i = 0$ and $\sigma_0 = \underline{\sigma}$;
repeat
    step 1 recover feasibility: given $\sigma_i$, find $z^i$ which is the solution to the LCP
        $0 \le z^i \perp q + \sigma_i p + (M + \sigma_i N)z^i \ge 0;$
    step 2 compute $(z^{i+0.5}, \sigma_{i+1})$ as the solution to the sub-problem
        minimize$_{\sigma,z}$ $f(\sigma, z) + \frac{c}{2}[(\sigma - \sigma_i)^2 + \|z - z^i\|^2]$
        subject to $\sigma \in [\underline{\sigma}, \bar{\sigma}]$,                                  (5.6)
        and $0 \le z \perp (q - \sigma_i N z^i) + \sigma(p + N z^i) + (M + \sigma_i N)z \ge 0;$
    set $i \leftarrow i + 1$;
until a stopping test is satisfied;
```

To solve the sequence of sub-problems (5.6) in Algorithm 10, we apply a method similar to the parametric pivoting method discussed in [Mur88, Chapter 5] and [CPS09, Chapter 4]. The fundamental idea of the method is to break the feasible range $\Gamma = [\underline{\sigma}, \bar{\sigma}]$ into sub-intervals (σ_i, σ_{i+1}), in which $z(\sigma)$ shares the same index sets α, β and γ for all $\sigma \in (\sigma_i, \sigma_{i+1})$. Here, the index sets α, β and γ associated with $z(\sigma)$ are defined as follows.

$$\begin{aligned}
\alpha &\equiv \{k : z(\sigma)_k > 0 = \big((q - \sigma_i N z^i) + \sigma(p + N z^i) + (M + \sigma_i N) z(\sigma)\big)_k\}, \\
\beta &\equiv \{k : z(\sigma)_k = 0 = \big((q - \sigma_i N z^i) + \sigma(p + N z^i) + (M + \sigma_i N) z(\sigma)\big)_k\}, \\
\gamma &\equiv \{k : z(\sigma)_k = 0 < \big((q - \sigma_i N z^i) + \sigma(p + N z^i) + (M + \sigma_i N) z(\sigma)\big)_k\}.
\end{aligned} \tag{5.7}$$

Especially, set α is called the active set. We restrict the sub-problem (5.6) on the sub-interval $[\sigma_i, \sigma_{i+1}]$ and use the fact that $z(\sigma)$ shares the same index sets for all σ in the sub-interval to simplify the sub-problem (5.6). For all z_k, $k \in \beta \cup \gamma$, we replace them by zero in the sub-problem (5.6). For all z_k, $k \in \alpha$, we express them in terms of σ in the sub-problem (5.6) by solving a system of linear equations. Therefore, we convert the sub-problem (5.6) into a one-dimensional mathematical program in the variable σ. Once we solve the sequence of mathematical programs for all sub-intervals, we compare the resulting optimal objective values in all sub-intervals. The point with the smallest optimal objective value is then considered as the optimal solution of the sub-problem (5.6). As suggested in [Mur88], we use a series of single principal pivot steps to break the feasible range Γ into sub-intervals. The pivot steps are easy to implement since the implementation only requires the knowledge of the associated index sets of each sub-interval. We now present our algorithm for solving the problem

$$\begin{aligned}
&0 \leq z \perp Mz + q + \sigma p \geq 0, \\
&\sigma \in [\underline{\sigma}, \bar{\sigma}]
\end{aligned} \tag{5.8}$$

in Algorithm 11. Interesting readers can find more discussions of the parametric pivoting method in [Mur88, Chapter 5].

5.3.2 Convergence analysis

Now we have presented our algorithm for solving the doubly uni-parametric MPCC in the case that the associated matrices are positive definite, we further develop the convergence result associated with this new algorithm. In this section, we begin with the introduction of some key properties related to function $z(\sigma)$ and the mathematical proofs of the properties. Based on the properties of $z(\sigma)$, we can derive our main convergence result that any non-degenerate limit point σ_* of the sequence $\{\sigma_i\}$ generated by Algorithm 10 is a B-stationary point of the objective function given that we choose the parameter $c > 0$ properly. Here, a point σ is non-degenerate if the associated index set $\beta = \{i \mid z(\sigma)_i = 0 \text{ and } ((M + \sigma N) z(\sigma) + (q + \sigma p))_i = 0\}$ of

Algorithm 11: Pivoting method for solving problem (5.8)

```
Input: a positive definite matrices $M \in \mathbf{R}^{n\times n}$, vectors $q \in \mathbf{R}^n$ and $p \in \mathbf{R}^n$;
set $i = 0$ and $\sigma_0 = \underline{\sigma}$;
solve for $z^0 \in SOL(q + \sigma_0 p, M)$;
determine the active set $\alpha$ associated with $z^0$;
repeat
    step 1 determine the range of $\sigma$, such that $z(\sigma)$ shares the same active set $\alpha$ for all $\sigma$ in the range:
    calculate the present complementary basis $\tilde{M}$ by
        $$\tilde{M}_{lk} = \begin{cases} M_{lk}, & l \in \alpha, \\ -1, & l \in \bar{\alpha} \text{ and } l = k, \\ 0, & otherwise; \end{cases}$$
    update the constant vectors
        $$q^* = \tilde{M}^{-1} q,$$
        $$p^* = \tilde{M}^{-1} p;$$
    calculate the feasible range $[\sigma_i, \sigma_{i+1}]$ by
        $$\sigma_{i+1} = \min\{\min_l\{\frac{-q_l^*}{p_l^*} : p_l^* > 0\}, \bar{\sigma}\};$$
    step 2 determine $z(\sigma)$ for all $\sigma \in [\sigma_i, \sigma_{i+1})$ by
        $$z_l = \begin{cases} -q_l^* - \sigma p_l^*, & l \in \alpha, \\ 0, & otherwise; \end{cases}$$
    step 3
    if $\sigma_{i+1} < \bar{\sigma}$ then
        choose $r = \max_l\{\frac{-q_l^*}{p_l^*} = \sigma_{i+1} : p_l^* > 0\}$;
        update the associated active set $\alpha$ by
            $$\alpha = \begin{cases} \alpha \cup \{r\}, & r \notin \alpha, \\ \alpha \setminus \{r\}, & otherwise; \end{cases}$$
        set $i \leftarrow i + 1$;
    else
        break;
until $\sigma_i = \bar{\sigma}$;
```

$z(\sigma)$ is empty. It is worth mentioning that the non-degeneracy of the limit point lays the foundation of our main convergence result. Another important concept in our main result is the concept of B-stationarity. The definition of B-stationarity is derived based upon the directional derivatives of $z(\sigma)$, which are themselves defined by

$$dz^{\pm}(\sigma) = \lim_{d\sigma \downarrow 0} \frac{z(\sigma \pm d\sigma) - z(\sigma)}{d\sigma}.$$

Hence, for an optimization problem

$$\text{minimize}_{\sigma \in \Gamma}\; g(\sigma) = f(\sigma, z(\sigma)),$$

its directional derivatives are calculated as

$$\begin{aligned} dg^{\pm}(\sigma) &= \lim_{d\sigma \downarrow 0} \frac{g(\sigma \pm d\sigma) - g(\sigma)}{d\sigma} \\ &= \pm \partial_\sigma f\left(\sigma, z(\sigma)\right) + \bigtriangledown_z f\left(\sigma, z(\sigma)\right)^T dz^{\pm}(\sigma). \end{aligned}$$

Let *int* Γ be the interior of set Γ. Then a scalar $\sigma \in int\,\Gamma$ is a B-Stationary point if the two one-sided directional derivatives $dg^{\pm}(\sigma) \geq 0$. Only one of the two one-sided derivatives is required to be non-negative if σ is one of the two end points of Γ.

We now derive the properties of $z(\sigma)$ for the case that the matrices M and N are both positive definite. Our main result comes naturally from those properties of $z(\sigma)$.

Lemma 5.3.1. *Let $z(\sigma)$ be the unique solution to the $LCP(q + \sigma p, M + \sigma N)$ for any given positive definite matrices M and N. Then, it follows that $z(\sigma)$ is a continuous function for $\sigma \geq 0$ and $\{z(\sigma) : \sigma \in [\underline{\sigma}, \bar{\sigma}]\}$ is bounded for $0 < \underline{\sigma} \leq \bar{\sigma} < \infty$.*

Proof. Since the matrices M and N are positive definite, $M + \sigma N$ is also positive definite and hence, a P-matrix. According to Theorem 2.2.2, we have that $z(\sigma)$ is the unique solution to the $LCP(q + \sigma p, M + \sigma N)$ and

$$z(\sigma)^T \left((M + \sigma N) z(\sigma) + (q + \sigma p)\right) = 0.$$

This is equivalent to

$$\begin{aligned} z(\sigma)^T (q + \sigma p) &= -z(\sigma)^T \left(M + \sigma N\right) z(\sigma) \\ &= -z(\sigma)^T \left(\frac{M + M^T}{2}\right) z(\sigma) - \sigma z(\sigma)^T \left(\frac{N + N^T}{2}\right) z(\sigma). \end{aligned} \tag{5.9}$$

Since M and N are positive definite matrices, we know that $\frac{M+M^T}{2}$ and $\frac{N+N^T}{2}$ are symmetric positive definite matrices. Therefore, using Weyl's inequality we know that the absolute value of the right-hand side of equation (5.9) is bounded below by

$$\left(\lambda_{\min}(\frac{M+M^T}{2})+\sigma\lambda_{\min}(\frac{N+N^T}{2})\right)\|z(\sigma)\|^2.$$

Here, $\lambda_{\min}(\cdot)$ is the smallest eigenvalue of a symmetric positive definite matrix. If we let

$$\lambda=\lambda_{\min}(\frac{M+M^T}{2})+\underline{\sigma}\lambda_{\min}(\frac{N+N^T}{2})>0,$$

then from (5.9) we have

$$\|z(\sigma)\|\|q+\sigma p\|\geq\lambda\|z(\sigma)\|^2.$$

Hence, $\|z(\sigma)\|$ is bounded above by $z_{\max}\equiv\frac{1}{\lambda}(\|q\|+\bar{\sigma}\|p\|)$ for any $\sigma\in[\underline{\sigma},\bar{\sigma}]$.

Now we have proved the boundedness of $z(\sigma)$ in the feasible range Γ, we further show that $z(\sigma)$ is a continuous function of σ using this result. For any σ and $\Delta\sigma$, we have

$$\begin{aligned}&0\leq z(\sigma)\perp(M+\sigma N)z+(q+\sigma p)\geq 0,\\&0\leq z(\sigma+\Delta\sigma)\perp\left(M+(\sigma+\Delta\sigma)N\right)z(\sigma+\Delta\sigma)+(q+(\sigma+\Delta\sigma)p)\geq 0.\end{aligned}$$

Therefore, we have

$$0\geq\left(z(\sigma+\Delta\sigma)-z(\sigma)\right)^T\left(\Delta\sigma p+\left(M+(\sigma+\Delta\sigma)N\right)z(\sigma+\Delta\sigma)-(M+\sigma N)z(\sigma)\right),$$

which is equivalent to

$$-\Delta\sigma(z(\sigma+\Delta\sigma)-z(\sigma))^Tp\geq\left(z(\sigma+\Delta\sigma)-z(\sigma)\right)^T\left((M+(\sigma+\Delta\sigma)N)z(\sigma+\Delta\sigma)-(M+\sigma N)z(\sigma)\right). \quad (5.10)$$

Let $A=M+\sigma N$ and $\widehat{A}=M+(\sigma+\Delta\sigma)N$, we have

$$\widehat{A}-A=\Delta\sigma N.$$

Then the right-hand side of inequality (5.10) equals

$$
\begin{aligned}
&(z(\sigma+\Delta\sigma)-z(\sigma))^T\left((\widehat{A}-A)(z(\sigma+\Delta\sigma)-z(\sigma))+\widehat{A}z(\sigma)+Az(\sigma+\Delta\sigma)-2Az(\sigma)\right)\\
&=(z(\sigma+\Delta\sigma)-z(\sigma))^T(\widehat{A}-A)(z(\sigma+\Delta\sigma)-z(\sigma))\\
&+(z(\sigma+\Delta\sigma)-z(\sigma))^T(\widehat{A}-A)z(\sigma)+(z(\sigma+\Delta\sigma)-z(\sigma))^TA(z(\sigma+\Delta\sigma)-z(\sigma))\\
&=\Delta\sigma(z(\sigma+\Delta\sigma)-z(\sigma))^TN(z(\sigma+\Delta\sigma)-z(\sigma))+\Delta\sigma(z(\sigma+\Delta\sigma)-z(\sigma))^TNz(\sigma)\\
&+(z(\sigma+\Delta\sigma)-z(\sigma))^TA(z(\sigma+\Delta\sigma)-z(\sigma)).
\end{aligned}
$$

Let $\Delta z = z(\sigma+\Delta\sigma)-z(\sigma)$, thus inequality (5.10) can be further written as

$$
-\Delta\sigma\Delta z^Tp \geq \Delta\sigma\Delta z^TN\Delta z+\Delta\sigma\Delta z^TNz(\sigma)+\Delta z^TA\Delta z. \tag{5.11}
$$

Rearrange inequality (5.11), we have

$$
-\Delta\sigma\Delta z^T(p+Nz(\sigma)) \geq \Delta\sigma\Delta z^TN\Delta z+\Delta z^TA\Delta z \geq 0. \tag{5.12}
$$

By taking the norms of both sides of (5.12), we have

$$
\begin{aligned}
|\Delta\sigma|\ \|\Delta z\|\ \|p+Nz(\sigma)\| &\geq \left(\lambda_{\min}(\frac{A+A^T}{2})+\Delta\sigma\lambda_{\min}(\frac{N+N^T}{2})\right)\|\Delta z\|^2\\
&\geq \left(\lambda_{\min}(\frac{A+A^T}{2})+\Delta\sigma\lambda_{\min}(\frac{N+N^T}{2})\right)\|\Delta z\|^2.
\end{aligned}
$$

Since the matrix A is the sum of the positive definite matrices M and σN, we have $\lambda_{\min}(\frac{A+A^T}{2}) \geq \lambda_{\min}(M) > 0$. Therefore, as $\Delta\sigma \to 0$,

$$
\|\Delta z\| \leq \frac{\|p+Nz(\sigma)\|}{\lambda_{\min}(\frac{M+M^T}{2})}|\Delta\sigma|.
$$

As shown above that $z(\sigma)$ is bounded for all $\sigma \in [\underline{\sigma},\bar{\sigma}]$, we have $\|\Delta z\| \leq \mathit{const} \times |\Delta\sigma| \to 0$ as $\Delta\sigma \to 0$. Therefore, $z(\sigma)$ is a continuous function of σ. □

In Lemma 5.3.1 , we have shown that for all $\sigma \in [\underline{\sigma},\bar{\sigma}]$, the solution $z(\sigma)$ of the $LCP(q+\sigma p, M+\sigma N)$ is a continuous and bounded function of σ given that matrices M and N are positive definite. Under the same condition that matrices M and N are positive definite, we can also show that there exists a common $l>0$ for all $\sigma \in [\underline{\sigma},\bar{\sigma}]$ such that the solution $z(q)$ to the $LCP(q, M+\sigma N)$ is a Lipschitz function of q with Lipschitz constant l via a similar approach.

Lemma 5.3.2. *For any given positive definite matrices M and N, there exists $l > 0$ such that for all $\sigma \in [\underline{\sigma}, \bar{\sigma}]$, if z and $\widehat{z}$ satisfy*

$$\begin{aligned} &0 \leq z \perp q + (M + \sigma N)z \geq 0, \\ &0 \leq \widehat{z} \perp \widehat{q} + (M + \sigma N)\widehat{z} \geq 0, \end{aligned} \tag{5.13}$$

then $\|z - \widehat{z}\| \leq l\|q - \widehat{q}\|$.

Proof. Since z and $\widehat{z}$ are the solutions to the LCPs in (5.13), we have

$$(z - \widehat{z})^T \left(q + (M + \sigma N)z - \widehat{q} - (M + \sigma N)\widehat{z}\right) \leq 0. \tag{5.14}$$

Rearranging inequality (5.14) leads to

$$\begin{aligned} -(z - \widehat{z})^T(q - \widehat{q}) \geq &(z - \widehat{z})^T(M + \sigma N)(z - \widehat{z}) \\ = &(z - \widehat{z})^T M(z - \widehat{z}) + \sigma(z - \widehat{z})^T N(z - \widehat{z}) \\ \geq &\lambda_{\min}\left(\frac{M + M^T}{2}\right)\|z - \widehat{z}\|^2 + \underline{\sigma}\lambda_{\min}\left(\frac{N + N^T}{2}\right)\|z - \widehat{z}\|^2. \end{aligned}$$

Let $\lambda = \lambda_{\min}(\frac{M+M^T}{2}) + \underline{\sigma}\lambda_{\min}(\frac{N+N^T}{2}) > 0$ and $l = \frac{1}{\lambda}$, we therefore have

$$\begin{aligned} &\lambda\|z - \widehat{z}\|^2 \leq \|q - \widehat{q}\|\ \|z - \widehat{z}\|, \\ \Longleftrightarrow\ &\|z - \widehat{z}\| \leq l\|q - \widehat{q}\|. \end{aligned}$$

□

Lemma 5.3.1 and Lemma 5.3.2 summarize the properties of z as a function of σ and q. Together with the boundedness, the continuity of z as a function of σ and the Lipschitz continuity of z as a function of q, we can show that the objective function is decreasing at each step if the parameter c is properly chosen. If in addition, the objective function is bounded below, we can further show that the sequence $\{z^i\}$ generated by Algorithm 10 converges by using the decreasing property of the objective function. The proof of the decreasing property of the objective function and the convergence of the sequence $\{z^i\}$ is quite straightforward and we present it in the following theorem:

Theorem 5.3.3. *If the objective function f is Lipschitz continuous with a Lipschitz constant $L > 0$, and the parameter c in Algorithm 10 is greater than $\frac{L\|N\|}{\lambda} > 0$, then the objective function will decrease at each step; moreover, if the objective function f is bounded below, then* $\lim_{i \to +\infty} z^i \to z^*$.

Proof. We first show that the objective function is decreasing at each step when $c > \frac{L\|N\|}{\lambda}$. Note that the difference of the objective values for two consecutive steps i and $i+1$ can be expressed as

$$\begin{aligned} & f(\sigma_{i+1}, z^{i+1}) - f(\sigma_i, z^i) \\ & = f(\sigma_{i+1}, z^{i+1}) - f(\sigma_{i+1}, z^{i+0.5}) + f(\sigma_{i+1}, z^{i+0.5}) - f(\sigma_i, z^i). \end{aligned} \tag{5.15}$$

Since $(\sigma_{i+1}, z^{i+0.5})$ is the optimal solution to the sub-problem (5.6), we have

$$f(\sigma_{i+1}, z^{i+0.5}) + \frac{c}{2}\left(\|z^{i+0.5} - z^i\|^2 + |\sigma_{i+1} - \sigma_i|^2\right) \leq f(\sigma_i, z^i). \tag{5.16}$$

Via the rearrangement and the substitution of (5.16) into (5.15), we obtain the following inequality:

$$\begin{aligned} & f(\sigma_{i+1}, z^{i+1}) - f(\sigma_i, z^i) \\ \leq\ & f(\sigma_{i+1}, z^{i+1}) - f(\sigma_{i+1}, z^{i+0.5}) - \frac{c}{2}\left(\|z^{i+0.5} - z^i\|^2 + |\sigma_{i+1} - \sigma_i|^2\right) \\ \leq\ & L\|z^{i+1} - z^{i+0.5}\| - \frac{c}{2}\left(\|z^{i+0.5} - z^i\|^2 + |\sigma_{i+1} - \sigma_i|^2\right). \end{aligned} \tag{5.17}$$

In addition, we also have

$$\begin{aligned} & 0 \leq z^{i+0.5} \perp (q + \sigma_{i+1}p) + (M + \sigma_{i+1}N)z^{i+0.5} - (\sigma_{i+1} - \sigma_i)N(z^{i+0.5} - z^i) \geq 0, \\ & 0 \leq z^{i+1} \perp (q + \sigma_{i+1}p) + (M + \sigma_{i+1}N)z^{i+1} \geq 0. \end{aligned}$$

Applying the result of Lemma 5.3.2 to the above two LCPs, we have that the difference between z^{i+1} and $z^{i+0.5}$ is bounded above by

$$\frac{1}{\lambda}|\sigma_{i+1} - \sigma_i|\|N\|\|z^{i+0.5} - z^i\|. \tag{5.18}$$

Combining inequality (5.17) and condition (5.18) together, we have

$$\begin{aligned} & f(\sigma_{i+1}, z^{i+1}) - f(\sigma_i, z^i) \\ \leq\ & \frac{L\|N\|}{\lambda}|\sigma_{i+1} - \sigma_i|\|z^{i+0.5} - z^i\| - \frac{c}{2}\left(|\sigma_{i+1} - \sigma_i|^2 + \|z^{i+0.5} - z^i\|^2\right) \\ \leq\ & \left(\frac{L\|N\|}{\lambda} - c\right)|\sigma_{i+1} - \sigma_i|\|z^{i+0.5} - z^i\|. \end{aligned} \tag{5.19}$$

Since $c > \frac{L\|N\|}{\lambda}$, the following condition holds:

$$f(\sigma_{i+1}, z^{i+1}) - f(\sigma_i, z^i) < 0.$$

Therefore, $f(\sigma_i, z^i)$ decreases at each step.

Next, we show that if in addition the objective function f is bounded below, then sequence $\{z^i\}$ converges. Recall in the monotone convergence theorem, we have that for any monotone sequence of real numbers, the sequence has a finite limit if and only if it is bounded. Therefore, due to the decreasing property and the boundedness of the objective function f, the limit of $\{f(\sigma_i, z^i)\}$ exists and

$$\lim_{i \to +\infty} f(\sigma_{i+1}, z^{i+1}) - f(\sigma_i, z^i) = 0. \tag{5.20}$$

Moreover, according to (5.19), we also have

$$|f(\sigma_{i+1}, z^{i+1}) - f(\sigma_i, z^i)| \geq \left(c - \frac{L\|N\|}{\lambda}\right) |\sigma_{i+1} - \sigma_i| \, \|z^{i+0.5} - z^i\|. \tag{5.21}$$

Combining (5.20) and (5.21) together, we have either

$$\begin{aligned} &(i) \lim_{i \to +\infty} \sigma_{i+1} - \sigma_i = 0, \text{ or} \\ &(ii) \lim_{i \to +\infty} z^{i+0.5} - z^i = 0. \end{aligned}$$

If $\lim_{i \to +\infty} \sigma_{i+1} - \sigma_i = 0$, then $\lim_{i \to +\infty} z^{i+1} - z^i = 0$ follows naturally due to the continuity of $z(\sigma)$ as a function of σ. If $\lim_{i \to +\infty} z^{i+0.5} - z^i = 0$, then according to (5.18) we have

$$\lim_{i \to +\infty} z^{i+1} - z^{i+0.5} = 0.$$

Therefore,

$$\lim_{i \to +\infty} z^{i+1} - z^i = \left(\lim_{i \to +\infty} z^{i+1} - z^{i+0.5}\right) + \left(\lim_{i \to +\infty} z^{i+0.5} - z^i\right) = 0,$$

and the limit of sequence $\{z^i\}$ exists. We denote $z^* \equiv \lim_{i \to +\infty} z^i$. □

The proof of the decreasing objective function and the convergence of the sequence $\{z^i\}$ relies on the fact that the change of the objective value $f(\sigma_i)$ is bounded linearly by the change of σ_i and the difference between $z^{i+0.5}$ and z^i. The feasibility of the limit point z^* follows naturally. Consider any convergent subsequence of $\{\sigma_i\}$ satisfying $\lim_{i_k \to +\infty} \sigma_{i_k} = \sigma_*$, we must have

$$\lim_{i_k \to +\infty} z(\sigma_{i_k}) = z(\sigma_*) = z^*.$$

Therefore, z^* is the unique solution to the $LCP(q + \sigma^* p, M + \sigma^* N)$.

Next, we focus on the limit point σ_* of a convergence subsequence: $\{\sigma_{i_k}\}$, and study the local properties of $z(\sigma_*)$ associated with the index sets for the final establishment of the convergence result. For a better demonstration of the local properties of $z(\sigma_*)$, we first introduce some notations. We know that for each σ, $z(\sigma)$ is the unique solution to the $LCP(q + \sigma p, M + \sigma N)$. Hence, the active set α is well-defined for each $z(\sigma)$ and can be viewed as a function of σ: $\alpha(\sigma)$. For a given set $\alpha \subseteq \{1, 2, ..., n\}$, we define $S_\alpha = \{\sigma : \alpha \text{ is the active set of } z(\sigma)\}$. According to the definition of S_α, it is easy to verify that

$$\bigcup_{\alpha \subseteq \{1,2,\ldots,n\}} S_\alpha = [\underline{\sigma}, \bar{\sigma}],$$
$$S_{\alpha^i} \bigcap S_{\alpha^j} = \{\emptyset\}, \ \forall \alpha^i, \alpha^j \subseteq \{1, 2, ..., n\} \text{ and } \alpha^i \neq \alpha^j.$$

The active set $\alpha(\sigma)$ of $z(\sigma)$ plays an important role in the local properties of $z(\sigma)$ when σ undergoes small perturbations. In the following two lemmas, Lemma 5.3.4 and Lemma 5.3.6, we present two local properties of $z(\sigma)$ associated with the active set that are closely related to our convergence result. Moreover, Lemma 5.3.6 can be viewed as an important consequence of Lemma 5.3.4.

Lemma 5.3.4. *For a given $\alpha \subseteq \{1, 2, ..., n\}$ and any positive definite matrices M and N, S_α is a finite union of sub-intervals of $\Gamma = [\underline{\sigma}, \bar{\sigma}]$; moreover, for any $\sigma \in \Gamma$ that does not belong to a finite set of points, the active set of $z(\sigma)$ remains unchanged locally.*

Proof. For a given $\alpha \subseteq \{1, 2, ..., n\}$ and any $z(\sigma)$ having α as its active set, the following conditions hold:

$$\begin{aligned} &- (M + \sigma N)^{-1}_{\alpha\alpha}(q + \sigma p) = z_\alpha(\sigma) > 0, \\ &- (M + \sigma N)_{\bar{\alpha}\alpha}(M + \sigma N)^{-1}_{\alpha\alpha}(q + \sigma p)_\alpha + (q + \sigma p)_{\bar{\alpha}} \geq 0 \end{aligned} \tag{5.22}$$

Since $M + \sigma N$ is a positive definite matrix, $(M + \sigma N)_{\alpha\alpha}$ is invertible for any α. Furthermore, all entries of $(M + \sigma N)^{-1}_{\alpha\alpha}$ can be viewed as rational functions of σ. Therefore, the two left-hand side expressions of (5.22) are well-defined rational functions of σ. The solutions to the above inequalities (5.22) are finite unions of sub-intervals of Γ. Hence, other than a finite number of points that are the isolated solutions to inequalities (5.22), all σ in each sub-interval of Γ share the same active set α. □

The piece-wise smoothness of $z(\sigma)$, whose definition is defined below, follows directly from Lemma 5.3.4.

Definition 5.3.5 (piece-wise smoothness). A real-valued function g defined on a nonempty open set $\Omega \subseteq \mathbf{R}^n$ is piece-wise smooth on Ω if it is continuous on Ω and there exists a finite collection of smooth functions g_i: $\Omega \to \mathbf{R}$, $i = 1, 2, \ldots, m$, such that

$$g(x) \in \{g_i(x) : i \in \{1, 2, \ldots, m\}\},\ \forall x \in \Omega.$$

Such a collection (not necessarily unique) is called a representation of g on Ω.

Lemma 5.3.6. *The solution $z(\sigma)$ to the LCP$(q + \sigma p, M + \sigma N)$ is a piece-wise smooth function of σ given that the matrices M and N are positive definite.*

Proof. Following Lemma 5.3.4, we have that the active set $\alpha(\sigma)$ of $z(\sigma)$ remains unchanged in an interval $\Theta(\sigma)$ for any $\sigma \in [\underline{\sigma}, \bar{\sigma}]$ that does not belong to a finite set of points. Given any such σ_0 and the corresponding $\Theta(\sigma_0)$, we hence have

$$\begin{aligned} z_{\alpha(\sigma_0)}(\sigma) &= -(M + \sigma N)^{-1}_{\alpha(\sigma_0)\alpha(\sigma_0)}(q + \sigma p), \\ z_{\bar{\alpha}(\sigma_0)}(\sigma) &= 0, \end{aligned}$$

for all $\sigma \in \Theta(\sigma_0)$. By solving the above system of equations, we can express $z(\sigma)$ as an analytical function of σ in the interval $\Theta(\sigma_0)$. Therefore, the function $z(\sigma)$ is smooth in the interval $\Theta(\sigma_0)$. Since there are only a finite number of subsets for the finite set $\{1, 2, \ldots, n\}$, there are a finite number of $S_\alpha, \alpha \subseteq \{1, \ldots, n\}$. Moreover, according to the result of Lemma 5.3.4, each S_α is a finite union of sub-intervals of $[\underline{\sigma}, \bar{\sigma}]$. Hence, $[\underline{\sigma}, \bar{\sigma}]$ can be broken into a finite number of sub-intervals, in which $z(\sigma)$ is locally smooth. By the definition of the piece-wise smoothness, we have that the function $z(\sigma)$ is piece-wise smooth. □

Lemma 5.3.4 and Lemma 5.3.6 have several important consequences. One of them is that the local constancy of the active set $\alpha(\sigma)$ provides a way to calculate $z(\sigma)$ analytically. Additionally, if $z(\sigma)$ is a non-degenerate solution, we further show that its directional derivatives $dz^{\pm}(\sigma)$ can be expressed analytically by solving systems of linear equations. Our main convergence result will be built upon the analytical form of the the directional derivatives $dz^{\pm}(\sigma)$.

Theorem 5.3.7. *If for a limit point σ_* of the sequence $\{\sigma_i\}$, the solution $z(\sigma_*)$ to the LCP$(q+\sigma_*, M+\sigma_* N)$ is non-degenerate, and $\bigtriangledown_z f(\cdot, \cdot)$, $\partial_\sigma f(\cdot, \cdot)$ are both Lipschitz continuous functions, then this limit point is a B-stationary point of the objective function $f(\sigma, z(\sigma))$.*

Proof. Without loss of generality, we assume that $\sigma_* \in \mathit{int}\ \Gamma$. For simplicity, we denote the convergent subsequence of $\{\sigma_i\}$ satisfying $\lim_{i_k \to +\infty} \sigma_{i_k} = \sigma_*$ as $\{\sigma_i\}$. For iteration i, we know that $(\sigma_{i+1}, z^{i+0.5})$ is the solution to sub-problem

$$\begin{aligned}
&\text{minimize}_{\sigma,z}\ f(\sigma,z)+\frac{c}{2}[(\sigma-\sigma_i)^2+\|z-z^i\|^2],\\
&\text{subject to } \sigma\in[\underline{\sigma},\bar{\sigma}],\\
&\text{and } 0\le z\perp(q-\sigma_i Nz^i)+\sigma(p+Nz^i)+(M+\sigma_i N)z\ge 0.
\end{aligned}$$

Therefore, $(\sigma_{i+1}, z^{i+0.5})$ must satisfy the conditions below:

$$\nabla_z f(\sigma_{i+1},z^{i+0.5})^T dz^{\pm}(\sigma_{i+1})\pm\partial_\sigma f\left(\sigma_{i+1},z(\sigma_{i+0.5})\right)+c(z^{i+0.5}-z^i)dz^{\pm}(\sigma_{i+1})\pm c(\sigma_{i+1}-\sigma_i)\ge 0, \quad (5.23)$$

with $dz^{\pm}(\sigma_{i+1})$ being the solutions dz to the mixed LCPs ([CPS09, Theorem 7.4.2])

$$\begin{aligned}
&\pm(p+Nz^i)_{\alpha^{i+0.5}}+(M+\sigma_i N)_{\alpha^{i+0.5}\alpha^{i+0.5}}dz_{\alpha^{i+0.5}}+(M+\sigma_i N)_{\alpha^{i+0.5}\beta^{i+0.5}}dz_{\beta^{i+0.5}}=0,\\
&\pm(p+Nz^i)_{\beta^{i+0.5}}+(M+\sigma_i N)_{\beta^{i+0.5}\alpha^{i+0.5}}dz_{\alpha^{i+0.5}}+(M+\sigma_i N)_{\beta^{i+0.5}\beta^{i+0.5}}dz_{\beta^{i+0.5}}\ge 0,\\
&dz_{\beta^{i+0.5}}{}^T\left(\pm(p+Nz^i)_{\beta^{i+0.5}}+(M+\sigma_i N)_{\beta^{i+0.5}\alpha^{i+0.5}}dz_{\alpha^{i+0.5}}+(M+\sigma_i N)_{\beta^{i+0.5}\beta^{i+0.5}}dz_{\beta^{i+0.5}}\right)=0,\\
&dz_{\beta^{i+0.5}}\ge 0,\ \text{and } dz_{\gamma^{i+0.5}}=0.
\end{aligned} \quad (5.24)$$

Here, $\alpha^{i+0.5}, \beta^{i+0.5}$, and $\gamma^{i+0.5}$ are the index sets associated with $z^{i+0.5}$:

$$\begin{aligned}
\alpha^{i+0.5}&=\{k: z_k^{i+0.5}>0=\left((q-\sigma_i Nz^i)+\sigma_{i+1}(p+Nz^i)+(M+\sigma_i N)z^{i+0.5}\right)_k\},\\
\beta^{i+0.5}&=\{k: z_k^{i+0.5}=0=\left((q-\sigma_i Nz^i)+\sigma_{i+1}(p+Nz^i)+(M+\sigma_i N)z^{i+0.5}\right)_k\},\\
\gamma^{i+0.5}&=\{k: z_k^{i+0.5}=0<\left((q-\sigma_i Nz^i)+\sigma_{i+1}(p+Nz^i)+(M+\sigma_i N)z^{i+0.5}\right)_k\}.
\end{aligned}$$

Similarly, we have that $dz^{\pm}(\sigma_*)$ are the solutions dz to the following trivial LCPs given that $z(\sigma_*)$ is a non-degenerate solution to the $LCP(q+\sigma_* p, M+\sigma_* N)$.

$$\begin{aligned}
&\pm(p+Nz^*)_{\alpha^*}+(M+\sigma_* N)_{\alpha^*\alpha^*}dz_{\alpha^*}=0,\\
&dz_{\gamma^*}=0.
\end{aligned} \quad (5.25)$$

Now that we have obtained the LCPs satisfied by both $dz^{\pm}(\sigma_{i+1})$ and $dz^{\pm}(\sigma_*)$, we study the relationship between the index sets associated with $z^{i+0.5}$ and z^*. If the index sets associated with $z^{i+0.5}$ and z^* are the same, then we obtain the following inequality:

$$\|dz^{\pm}(\sigma_{i+1})-dz^{\pm}(\sigma_*)\|\le \mathit{const}\times\left(|\sigma_i-\sigma_*|+\|z^i-z^*\|\right) \quad (5.26)$$

by directly applying the result of [CPS09, Lemma 7.3.10] to LCPs (5.24) and (5.25). Luckily, the index sets associated with $z^{i+0.5}$ and z^* are indeed the same when i is sufficiently large. The proof of the above claim is based on the direct application of Lemma 5.3.4 and presented as follows.

As $\sigma_i \to \sigma_*$, we have $z^i \to z^*$. By using the result of Lemma 5.3.4 that the active set $\alpha(\sigma_*)$ remains unchanged locally, we have that there exists $N_1 > 0$ such that for all $i \geq N_1$, z^i has the same active set as z^*. As shown in Theorem 5.3.3, we also have

$$\lim_{i \to +\infty} z^{i+0.5} - z^i = 0,$$
$$\lim_{i \to +\infty} z^{i+1} - z^{i+0.5} = 0.$$

Therefore, there exists $N_2 > 0$ such that for all $i > N_2$,

$$\alpha^i \supseteq \alpha^{i+0.5} \supseteq \alpha^{i+1},$$
$$\gamma^i \supseteq \gamma^{i+0.5} \supseteq \gamma^{i+1}.$$

Moreover,

$$\alpha^i = \alpha^{i+0.5} = \alpha^{i+1} = \alpha^*,$$
$$\gamma^i = \gamma^{i+0.5} = \gamma^{i+1} = \gamma^*,$$

for all $i > N \equiv \max(N_1, N_2)$. Thus, inequalities (5.26) hold and as $n \to \infty$, we have $dz^{\pm}(\sigma_{i+1}) \to dz^{\pm}(\sigma_*)$.

Besides, we also have that $\bigtriangledown_z f(\cdot, \cdot)$ and $\partial_\sigma f(\cdot, \cdot)$ are continuous Lipschitz functions. Accordingly,

$$\begin{aligned} | \bigtriangledown_z f(\sigma_{i+1}, z^{i+0.5}) - \bigtriangledown_z f(\sigma_*, z^*)| &\leq L_{\bigtriangledown_z f} \left(\|z^{i+0.5} - z^*\| + |\sigma_{i+1} - \sigma_*| \right), \\ |\partial_\sigma f(\sigma_{i+1}, z^{i+0.5}) - \partial_\sigma f(\sigma_*, z^*)| &\leq L_{\partial_\sigma f} \left(\|z^{i+0.5} - z^*\| + |\sigma_{i+1} - \sigma_*| \right). \end{aligned} \tag{5.27}$$

Here, $L_{\bigtriangledown_z f}$ and $L_{\partial_\sigma f}$ are the Lipschitz constants of $\bigtriangledown_z f(\cdot, \cdot)$ and $\partial_\sigma f(\cdot, \cdot)$, respectively. Let $n \to \infty$ for both sides of (5.23) and combine it with inequalities (5.27), we have

$$\bigtriangledown_z f(\sigma_*, z^*)^T dz^{\pm}(\sigma_*) \pm \partial_\sigma f(\sigma_*, z(\sigma_*) \geq 0.$$

Hence, we have proved that any non-degenerate limit point σ_* of the sequence $\{\sigma_i\}$ is a B-stationary point of the original problem (5.1). □

A special case of Theorem 5.3.7 occurs when there is no further decrease in the objective values after a few iterations. More precisely, we have that there exists $i > 0$ such that

$$f(z^{i+1}, \sigma_{i+1}) = f(z^i, \sigma_i). \tag{5.28}$$

Due to inequality (5.21), condition (5.28) implies

$$\begin{aligned} &|f(\sigma_{i+1}, z^{i+1}) - f(\sigma_i, z^i)| = 0 \geq \left(c - \frac{L\|N\|}{\lambda}\right) \; |\sigma_{i+1} - \sigma_i| \; \|z^{i+0.5} - z^i\| \\ \iff &\sigma_{i+1} = \sigma_i, \text{ or } z^{i+0.5} = z^i. \end{aligned}$$

If $\sigma_{i+1} = \sigma_i$, we have $z^{i+1} = z^i$ according to the uniqueness of the solution to a LCP with a positive definite matrix. It then follows naturally that $z^i = z^{i+0.5} = z^{i+1}$. If $z^{i+0.5} = z^i$, according to (5.18), we again have $z^i = z^{i+0.5} = z^{i+1}$. Therefore, $f(\sigma_{i+1}, z^{i+1}) = f(\sigma_i, z^i)$ implies $z^i = z^{i+0.5} = z^{i+1}$ in Algorithm 10.

The B-Stationarity of a limit point depends on the local constancy of the active set of the limit point. Especially when the limit point is non-degenerate, both $z(\sigma_*)$ and its derivatives $dz^{\pm}(\sigma_*)$ can be expressed analytically using the associated active set. Moreover, z^i, $z^{i+0.5}$ share the same active set with z^* when i is sufficiently large. Therefore, z^i, $z^{i+0.5}$ and their directional derivatives can be computed by solving systems of linear equations of similar forms. The B-Stationarity of the limit point comes naturally by taking the limit of the stationarity condition (5.23) of the sub-problems.

5.4 A new algorithm for the positive semi-definite case

We further extend Algorithm 10 to solve problem (5.1) with relaxed matrix constraints: positive semi-definite matrices M and N. The main challenges of this extension lie in the fact that with positive semi-definite matrices M and N, the solution $z(\sigma)$ is not necessarily unique for a given σ when LCP (5.4) is solvable. Therefore, modification to Algorithm 10 is needed to address the issue of the non-uniqueness of the solution set $SOL(q + \sigma p, M + \sigma p)$. We devote the rest of this section to the extension of Algorithm 10 for solving problem (5.1) in the case that the asscoiated matrices are positive semi-definite. This part of the work is arranged as follows. We first introduce the modified algorithm and its implementation details, then followed by the presentation of the theoretical results related with the convergence of the modified algorithm and their proofs.

5.4.1 Formal statement of the algorithm

The main focus of this section is to present a formal statement of the modified algorithm for solving problem (5.1) in case that the associated matrices M and N are positive semi-definite. The complete algorithm is summarized in Algorithm 12.

```
Algorithm 12: Modified linearized approach for an MPCC with positive semi-definite matrices
  Input: positive semi-definite matrices $M \in \mathbf{R}^{n\times n}$ and $N \in \mathbf{R}^{n\times n}$,
  vectors $q \in \mathbf{R}^n$ and $p \in \mathbf{R}^n$;
  set $i = 0$, $\sigma_0 = \underline{\sigma}$, and select an arbitrary $z^0$ from $SOL(q + \sigma_0 p, M + \sigma_0 N)$;
  repeat
      step 1 compute $(\sigma_{i+1}, z^{i+0.5})$ as the solution to the sub-problem

          $\text{minimize}_{\sigma,z}\ \ f(\sigma, z) + \frac{c}{2}[(\sigma - \sigma_i)^2 + \|z - z^i\|^2]$
          $\text{subject to}\ \ 0 \le z \perp (q - \sigma_i N z^i) + \sigma(p + N z^i) + (M + \sigma_i N)z \ge 0,$      (5.29)
          $\text{and}\ \sigma \in [\underline{\sigma}, \bar{\sigma}];$

      step 2 recover feasibility: given $\sigma_{i+1}$, find $z^{i+1}$ by solving the optimization problem

          $\text{minimize}_z\ \ \|z - z^{i+0.5}\|^2$
          $\text{subject to}\ \ 0 \le z \perp (q + \sigma_{i+1} p) + (M + \sigma_{i+1} N)z \ge 0;$      (5.30)

      set $i \leftarrow i + 1$;
  until a stopping test is satisfied;
```

In Algorithm 12, we initialize the method by setting $\sigma_0 = \underline{\sigma}$ and letting z^0 be an arbitrary solution to the $LCP(q + \sigma_0 p, M + \sigma_0 N)$. Similar to Algorithm 10, our modified algorithm is mainly composed of two steps at each iteration: (i) the computation of a search direction $d\sigma = \sigma_{i+1} - \sigma_i$; (ii) the recovery of the feasibility of the original complementarity constraints (5.4) at σ_{i+1}. The first step involves solving sub-problem (5.29), whose constraints are the linear approximations to constraints (5.4) at (σ_i, z^i). We again adopt a partition method to solve the sub-problem in the case of positive semi-definite matrices M and N. The validity of the partition method lies in one key property of function $z(\sigma)$.

To better demonstrate the key property of $z(\sigma)$, we first redefine a few concepts for the case that the associated matrices M and N are positive semi-definite. We redefine the index sets associated with z when the matrix M is positive semi-definite based on the fact that the solution set $SOL(q, M)$ is convex and has the following representation as a polyhedron:

$$SOL(q, M) = \{z \in \mathbf{R}^n_+ \mid (q + Mz)_\alpha = 0 \text{ and } (q + Mz)_{\bar{\alpha}} \ge 0\}.$$

Here, $\alpha = \alpha(q, M)$ is the redefined active set and given by

$$\alpha(q, M) \equiv \{l \mid \exists z \in SOL(q, M) \text{ with } z_l > 0\}, \tag{5.31}$$

and $\bar{\alpha}$ is the complement of α in $\{1, \ldots, n\}$. We then redefine the index sets β and γ by refining the above representation and partite the latter set $\bar{\alpha}$ into the disjoint union of two subsets of $\{1, \ldots, n\}$:

$$\begin{aligned} \beta = \beta(q, M) &\equiv \{j \mid \forall z \in SOL(q, M), z_j = 0 = (q + Mz)_j\}, \\ \gamma = \gamma(q, M) &\equiv \{k \mid \exists z \in SOL(q, M) \text{ with } (q + Mz)_k > 0\}. \end{aligned} \tag{5.32}$$

Any one of the three index sets $\alpha(q, M)$, $\beta(q, M)$ and $\gamma(q, M)$ may be empty, but their union must be equal to the full set $\{1, \ldots, n\}$. Other than the above redefinition of the index sets, the key property of $z(\sigma)$ also involves a new concept: cross complementarity defined by

Definition 5.4.1 (cross complementarity). For any two solutions $z^i \in LCP(q, M)$, $i \in \{1, 2\}$, we have $(z^1)^T(q + Mz^2) = 0$.

The cross complementarity property is naturally satisfied when the matrix M is positive semi-definite [CPS09, Proposition 3.1.8]. Given the redefinition of the index sets in case of positive semi-definite matrices and the definition of the cross complementarity, we now present the important property of $z(\sigma)$ upon which we build Algorithm 12 as follows.

Theorem 5.4.2. *Suppose that for all $\sigma \in \Gamma = [\underline{\sigma}, \bar{\sigma}]$, the cross complementarity property holds for the solutions of the $LCP(q + \sigma p, M)$, then there exists a finite partition of Γ:*

$$\underline{\sigma} \equiv \sigma_{-1} < \sigma_0 < \sigma_1 < \ldots < \sigma_{K-1} < \sigma_K < \sigma_{K+1} \equiv \bar{\sigma},$$

such that for all $k = 0, 1, \ldots, K+1$, the triplet of the index sets $(\alpha(q + \sigma p, M), \beta(q + \sigma p, M), \gamma(q + \sigma p, M))$ is a constant for $\sigma \in (\sigma_{k-1}, \sigma_k)$.

The proof of Theorem 5.4.2 can be found in a working paper by Pang. The consequence of Theorem 5.4.2 is that it provides a computationally effective way of solving sub-problem (5.29) with its associated matrix $M + \sigma_i N$ being positive semi-definite. By breaking the feasible range Γ into sub-intervals where the index sets remain unchanged in each sub-interval, we can solve the sub-problem in each sub-interval by transforming it into an optimization problem with linear constraints. The transformation is mainly based on the fact that the linear complementarity constraints with constant index sets are equivalent to linear constraints. We now present the partition algorithm in Algorithm 13. This algorithm is particularly designed for solving the following sub-problem (5.33) appeared in the first step of Algorithm 12:

$$\begin{aligned}
&\text{minimize}_{\sigma,z}\;\; f(\sigma,z)+\frac{c}{2}[(\sigma-\sigma_i)^2+\|z-z^i\|^2]\\
&\text{such that}\;\; 0\le z\perp(q-\sigma_i Nz^i)+\sigma(p+Nz^i)+(M+\sigma_i N)z\ge 0,\\
&\text{and}\;\sigma\in[\underline{\sigma},\bar{\sigma}].
\end{aligned}\tag{5.33}$$

It is worth mentioning that in step 3 of Algorithm 13, we solve the optimization problem (5.33) with σ fixed at the break point σ^{break}. By taking advantage of the polyhedral representation of $LCP(q-\sigma_i Nz^i+\sigma^{\text{break}}(p+Nz^i), M+\sigma_i N)$, the optimization problem (5.33) is equivalent to the following mathematical program with linear constraints [CPS09, Theorem 3.1.7c]:

$$\begin{aligned}
&\text{minimize}_{z\ge 0}\;\; f(\sigma^{\text{break}},z)+\frac{c}{2}\|z-z^i\|^2\\
&\text{subject to}\;\; \left(M_{\text{break}}+M_{\text{break}}^T\right)z=\left(M_{\text{break}}+M_{\text{break}}^T\right)z^{\text{break}},\\
&M_{\text{break}}z+q^{\text{break}}\ge 0,\\
&\text{and}\;\; (q^{\text{break}})^T z=(q^{\text{break}})^T z^{\text{break}},
\end{aligned}\tag{5.36}$$

with

$$\begin{aligned}
M_{\text{break}}&=M+\sigma_i N,\\
q^{\text{break}}&=q-\sigma_i Nz^i+\sigma^{\text{break}}(p+Nz^i),
\end{aligned}$$

and z^{break} being an arbitrary solution to the $LCP(q^{\text{break}}, M_{\text{break}})$. Existing solvers can be applied directly to solve optimization problems of form (5.36). Step 3 of Algorithm 13 is necessary because at each break point σ^{break}, $LCP(q-\sigma_i Nz^i+\sigma^{\text{break}}(p+Nz^i), M+\sigma_i N)$ might have multiple solutions with different index sets. Hence, it requires us to solve optimization problem (5.33) at σ^{break} separately. Notice that the overall complexity of Algorithm 13 is dominated by solving three sets of optimization problems with linear constraints: one set of linear programs to decide the break points, two sets of mathematical programs with linear constraints for determining the optimal solution of the sub-problem in each interval $[\sigma_i,\sigma_{i+1}]$.

In step 2 of Algorithm 12, we recover the feasibility of the solution (σ_{i+1}, z^{i+1}) by solving optimization problem

$$\begin{aligned}
&\text{minimize}_z\;\; \|z-z^{i+0.5}\|^2\\
&\text{subject to}\;\; 0\le z\perp(M+\sigma_{i+1}N)z+(q+\sigma_{i+1}p)\ge 0.
\end{aligned}$$

Again by taking advantage of the polyhedral representation of $SOL(q+\sigma_{i+1}p, M+\sigma_{i+1}N)$, we transform

Algorithm 13: Partition algorithm for solving sub-problem (5.33)

```
Input: positive semi-definite matrices $M \in \mathbf{R}^{n\times n}$ and $N \in \mathbf{R}^{n\times n}$,
vectors $q \in \mathbf{R}^n$ and $p \in \mathbf{R}^n$, and the pair $(\sigma_i, z^i)$;
solve problem (5.33) with $\sigma$ fixed at $\underline{\sigma}$;
record $\underline{\sigma}$, $z(\underline{\sigma})$ and the corresponding objective value $f(\underline{\sigma}, z(\underline{\sigma}))$ as $(\sigma_{\text{opt}}, z_{\text{opt}}, f_{\text{opt}})$;
Notice that solving problem (5.33) with $\sigma$ fixed at $\sigma_{\text{fix}}$ is equivalent to solving the nonlinear program

    $$\begin{aligned} &\text{minimize}_{z\geq 0}\ f(\sigma_{\text{fix}}, z) + \frac{c}{2}\|z - z^i\|^2 \\ &\text{subject to}\ \ (q - \sigma_i N z^i) + \sigma_{\text{fix}}(p + Nz^i) + (M + \sigma_i N)z \geq 0, \\ &[(M + \sigma_i N) + (M + \sigma_i N)^T]z = [(M + \sigma_i N) + (M + \sigma_i N)^T]z_{\text{fix}}, \\ &\text{and } [(q - \sigma_i N z^i) + \sigma_{\text{fix}}(p + Nz^i)]^T z = [(q - \sigma_i N z^i) + \sigma_{\text{fix}}(p + Nz^i)]^T z_{\text{fix}}, \end{aligned} \tag{5.34}$$

where $z_{\text{fix}}$ is an arbitrary solution of the $\mathit{SOL}\big((q - \sigma_i N z^i) + \sigma_{\text{fix}}(p + Nz^i), M + \sigma_i N\big)$;
set $\sigma^{\min} = \underline{\sigma}$ and define $z^{\min} \equiv z(\sigma^{\min})$;
repeat
    step 1 find $\sigma^{\max}$ such that $\forall \sigma \in (\sigma^{\min}, \sigma^{\max})$, $z(\sigma)$ shares the same index sets $\alpha$, $\beta$ and $\gamma$ with $z^{\min}$; this can be achieved by solving the linear program

        $$\begin{aligned} &\sigma^{\max} \equiv \text{maximize}_{\sigma,z}\sigma \\ &\text{subject to } \sigma \leq \bar{\sigma}, \\ &z_\alpha \geq 0 = \Big((q - \sigma_i N z^i) + \sigma(p + Nz^i) + (M + \sigma_i N)z\Big)_\alpha, \\ &z_\beta = 0 = \Big((q - \sigma_i N z^i) + \sigma(p + Nz^i) + (M + \sigma_i N)z\Big)_\beta, \\ &\text{and } z_\gamma = 0 \leq \Big((q - \sigma_i N z^i) + \sigma(p + Nz^i) + (M + \sigma_i N)z\Big)_\gamma; \end{aligned}$$

    step 2
    if $\sigma^{\max} > \bar{\sigma}$, then
        let $\sigma^{\max} = \bar{\sigma}$;
    solve problem (5.33) in interval $(\sigma^{\min}, \sigma^{\max})$;
    this can be achieved by solving the nonlinear program

        $$\begin{aligned} &\text{minimize}_{\sigma,z} f(\sigma, z) + \frac{c}{2}[(\sigma - \sigma_i)^2 + \|z - z^i\|^2] \\ &\text{subject to } \sigma^{\min} \leq \sigma \leq \sigma^{\max}, \\ &z_\alpha \geq 0 = \Big((q - \sigma_i N z^i) + \sigma(p + Nz^i) + (M + \sigma_i N)z\Big)_\alpha, \\ &z_\beta = 0 = \Big((q - \sigma_i N z^i) + \sigma(p + Nz^i) + (M + \sigma_i N)z\Big)_\beta, \\ &\text{and } z_\gamma = 0 \leq \Big((q - \sigma_i N z^i) + \sigma(p + Nz^i) + (M + \sigma_i N)z\Big)_\gamma; \end{aligned} \tag{5.35}$$

    update $(\sigma_{\text{opt}}, z_{\text{opt}}, f_{\text{opt}})$ by the following rule:
    if the objective value obtained from (5.35) is smaller than the current $f_{\text{opt}}$, then
        update $(\sigma_{\text{opt}}, z_{\text{opt}}, f_{\text{opt}})$ using the solution of (5.35) and the corresponding objective value;
    else
        $(\sigma_{\text{opt}}, z_{\text{opt}}, f_{\text{opt}})$ remains unchanged;
    step 3 solve problem (5.33) with $\sigma$ fixed at $\sigma^{\max}$;
    update $(\sigma_{\text{opt}}, z_{\text{opt}}, f_{\text{opt}})$ using the same rule discussed in step 2;
    step 4
    if $\sigma^{\max} = \bar{\sigma}$, then
        break;
    else
        let $\sigma^{\min} = \sigma^{\max}$;
        update $z^{\min}$ and the corresponding index sets $\alpha$, $\beta$ and $\gamma$ of $z^{\min}$;
until $\sigma^{\max} = \bar{\sigma}$;
```

the above optimization problem to quadratic program

$$\begin{aligned}
&\text{minimize}_{z\geq 0} \;\; \|z - z^{i+0.5}\|^2\\
&\text{subject to } \;\big(M + M^T + \sigma_{i+1}(N + N^T)\big)\, z = \big(M + M^T + \sigma_{i+1}(N + N^T)\big)\, z^{\text{fixed}},\\
&(M + \sigma_{i+1}N)z + (q + \sigma_{i+1}p) \geq 0,\\
&\text{and } \;(q + \sigma_{i+1}p)^T z = (q + \sigma_{i+1}p)^T z^{\text{fixed}}.
\end{aligned}$$

Here, $z^{\text{fixed}} \in SOL(q + \sigma_{i+1}p, M + \sigma_{i+1}N)$.

5.4.2 Convergence analysis

In this section, we present the convergence result of the modified algorithm 12 for solving problem (5.1) in the case that the matrices M and N are positive semi-definite. We show that under mild conditions, any non-degenerate limit point of the sequence generated by Algorithm 12 satisfies the first-order KKT conditions of problem (5.1). Before we provide the details of the convergence theorem and its proof, we first introduce some definitions related to our assumptions and properties of $\{z^i\}$, upon which we build our convergence result.

Definition 5.4.3 (R_0 matrix). A matrix M is an R_0 matrix if 0 is the unique solution to the LCP:

$$0 \leq z \perp Mz \geq 0.$$

The boundedness of sequence $\{z^i\}$ can be derived from the R_0 property of the matrices $M+\sigma N$, $\forall \sigma \in \Gamma$.

Lemma 5.4.4. *If $M+\sigma N$ is an R_0 matrix for any $\sigma \in [\underline{\sigma}, \bar{\sigma}]$, then the sequence $\{z^i\}$ generated by Algorithm 12 is bounded.*

Proof. Assume that $\{z^i\}$ is unbounded. Since $z^i \in SOL(q+\sigma_i p, M+\sigma_i N)$, z^i satisfies the following condition:

$$0 \leq z^i \perp (q + \sigma_i p) + (M + \sigma_i N)z^i \geq 0. \tag{5.37}$$

Dividing both sides of (5.37) by $\|z^i\|$ gives

$$0 \leq \frac{z^i}{\|z^i\|} \perp \frac{(q + \sigma_i p)}{\|z^i\|} + (M + \sigma_i N)\frac{z^i}{\|z^i\|} \geq 0. \tag{5.38}$$

Let $\widehat{z^i} \equiv \frac{z^i}{\|z^i\|}$, then $\widehat{z^i}$ is bounded with $\|\widehat{z^i}\| = 1$. Therefore, we can choose a convergent subsequence of $\{\sigma_i\}$ such that $\lim_{i_k\to+\infty} \sigma_{i_k} = \sigma_*$ and $\lim_{i_k\to+\infty} \widehat{z^{i_k}} \to \widehat{z^*}$ with $\|\widehat{z^*}\| = 1$. Since the sequence $\{z^{i_k}\}$ is not bounded, taking the limit of both sides of equation (5.38) leads to

$$0 \leq \widehat{z^*} \perp (M + \sigma_* N)\widehat{z^*} \geq 0.$$

Therefore $\widehat{z^*} \neq 0 \in SOL(0, M + \sigma_* N)$. This contradicts with the condition that $M + \sigma_* N$ is a R_0 matrix. Hence, $\{z^i\}$ is bounded. □

We have proved that the sequence $\{z^i\}$ generated by Algorithm 12 is bounded if the matrix $M + \sigma N$ is R_0 for all $\sigma \in [\underline{\sigma}, \bar{\sigma}]$. Other than the boundedness of $\{z^i\}$, we also study the relationship between z^i, $z^{i+0.5}$ and z^i in the following lemma:

Lemma 5.4.5. *At any iteration i of Algorithm 12, there exist positive constants ψ_i and κ_i such that*

$$\|z^{i+0.5} - z^i\| \leq \psi_i \left(\, |\sigma_{i+1} - \sigma_i| \left(\|p\| + \|N\| \, \|z^i\| \right) \right)^{\kappa_i}.$$

Proof. Applying [FP03a, Theorem 6.6.6] to the LCPs satisfied by $z^{i+0.5}$ and z^i leads to

$$dist\left((y^{i+0.5}, z^{i+0.5}), (y^i, z^i)\right) \leq \psi_i \widehat{\varpi}(y^{i+0.5}, z^{i+0.5})^{\kappa_i}.$$

Here, $\psi_i > 0$ and $\kappa_i > 0$, and

$$\begin{aligned} & y^{i+0.5} \equiv (q - \sigma_i N z^i) + \sigma_{i+1}(p + N z^i) + (M + \sigma_i N) z^{i+0.5}, \\ & y^i \equiv (q + \sigma_i p) + (M + \sigma_i N) z^i, \\ & \widehat{\varpi}(y, z) \equiv \|\widehat{H}(y, z)\| + \|y_-\| + \|z_-\| + |y^T z|, \\ & \widehat{H}(y, z) \equiv y - (q + \sigma_i p) - (M + \sigma_i N) z. \end{aligned}$$

Since $(\sigma_{i+1}, z^{i+0.5})$ is the optimal solution to problem (5.33), we have

$$\|y_-^{i+0.5}\| = \|z_-^{i+0.5}\| = (y^{i+0.5})^T z^{i+0.5} = 0.$$

Therefore,

$$\begin{aligned}\widehat{\varpi}(y^{i+0.5}, z^{i+0.5}) &= \|\widehat{H}(y^{i+0.5}, z^{i+0.5})\| \\ &= \|y^{i+0.5} - (q + \sigma_i p) - (M + \sigma_i N) z^{i+0.5}\| \\ &= \|(q - \sigma_i N z^i) + \sigma_{i+1}(p + N z^i) + (M + \sigma_i N) z^{i+0.5} - (M + \sigma_i N) z^{i+0.5} - (q + \sigma_i p)\| \\ &= |\sigma_{i+1} - \sigma_i| \ \left(\|p + N z^i\|\right).\end{aligned}$$

Hence,

$$\begin{aligned}\|z^{i+0.5} - z^i\| \leq & dist\left((y^{i+0.5}, z^{i+0.5}), (y^i, z^i)\right) \\ \leq & \psi_i \left(|\sigma_{i+1} - \sigma_i| \ \left(\|p\| + \|N\| \ \|z^i\|\right)\right)^{\kappa_i}.\end{aligned}$$

□

Next, we show that the objective function f is decreasing under certain mild conditions by applying a similar technique used in the proof of Theorem 5.3.3. The convergence of the sequence $\{z^i\}$ follows naturally from the decreasing property of the objective function f.

Theorem 5.4.6. *If in Algorithm 12, there exist (i)* $G > 0$ *such that* $\|z^{i+1} - z^{i+0.5}\| < G\,|\sigma_{i+1} - \sigma_i| \|z^{i+0.5} - z^i\|$ *for all* i*; (ii) positive* ψ *and* κ *such that for all* ψ_i *and* κ_i*,* $\psi_i \leq \psi$ *and* $\kappa_i \leq \kappa$*. If in addition, the objective function* f *is Lipschitz continuous with Lipschitz constant* $L > 0$ *and the parameter* $c > LG > 0$*, then the objective function will decrease at each step; furthermore, if the objective function* f *is bounded below, then* $\lim_{i \to +\infty} z^i \to z^*$.

Proof. Using (5.17) from the proof of Theorem 5.3.3, we have that the difference between the objective values at two consecutive steps i and $i+1$ satisfies the following condition:

$$\begin{aligned}& f(\sigma_{i+1}, z^{i+1}) - f(\sigma_i, z^i) \\ \leq \ & L\|z^{i+1} - z^{i+0.5}\| - \frac{c}{2}\left(\|z^{i+0.5} - z^i\|^2 + |\sigma_{i+1} - \sigma_i|^2\right).\end{aligned}$$

Under the assumption that

$$\|z^{i+1} - z^{i+0.5}\| < G|\sigma_{i+1} - \sigma_i| \|z^{i+0.5} - z^i\|, \tag{5.39}$$

we have

$$\begin{aligned}& f(\sigma_{i+1}, z^{i+1}) - f(\sigma_i, z^i) \\ \leq \ & (LG - c)|\sigma_{i+1} - \sigma_i| \ \|z^{i+0.5} - z^i\|.\end{aligned}$$

Since $c > LG$, we have

$$f(\sigma_{i+1}, z^{i+1}) - f(\sigma_i, z^i) < 0.$$

Therefore, $f(\sigma_i, z^i)$ decreases at each step.

Similar to the proof of Theorem 5.3.3, we show the convergence of the sequence $\{z^i\}$ by using the decreasing and boundedness property of the objective function f. Again by the monotone convergence theorem, we have

$$\lim_{i\to+\infty} f(\sigma_{i+1}, z^{i+1}) - f(\sigma_i, z^i) = 0.$$

Since $|f(\sigma_{i+1}, z^{i+1}) - f(\sigma_i, z^i)| \geq (c - LG)\ |\sigma_{i+1} - \sigma_i|\ \|z^{i+0.5} - z^i\|$, either

$$\begin{aligned} &(i)\ \lim_{i\to+\infty}\ \sigma_{i+1} - \sigma_i = 0, \text{ or} \\ &(ii)\ \lim_{i\to+\infty}\ z^{i+0.5} - z^i = 0 \end{aligned}$$

holds. As shown in Lemma 5.4.5,

$$\begin{aligned} \|z^{i+0.5} - z^i\| &\leq \psi_i \left(|\sigma_{i+1} - \sigma_i| \left(\|p\| + \|N\|\|z^i\|\right)\right)^{\kappa_i} \\ &\leq \psi \left(|\sigma_{i+1} - \sigma_i| \left(\|p\| + \|N\|\|z^i\|\right)\right)^{\kappa}. \end{aligned} \tag{5.40}$$

Notice that $\lim_{i\to+\infty}\ \sigma_{i+1} - \sigma_i = 0$ leads to $\lim_{i\to+\infty}\ z^{i+0.5} - z^i = 0$ according to (5.40). Given $\lim_{i\to+\infty}\ z^{i+0.5} - z^i = 0$, we have

$$\lim_{i\to+\infty} z^{i+1} - z^{i+0.5} = 0$$

under the assumption (5.39). Therefore,

$$\lim_{i\to+\infty} z^{i+1} - z^i = 0.$$

Denote $\lim_{i\to+\infty} z^i$ as z^*, we have shown that the sequence $\{z^i\}$ converges to z^*.

□

Similar to the positive definite case, we can derive the feasibility of the limit point z^*. For any convergent subsequence of $\{\sigma_i\}$ satisfying $\lim_{i_k\to+\infty} \sigma_{i_k} = \sigma_*$, we must have

$$\lim_{i_k\to+\infty} z(\sigma_{i_k}) = z(\sigma^*) = z^*.$$

We now focus on the limit point σ_* of a convergence subsequence $\{\sigma_{i_k}\}$, and study the local properties of $z(\sigma_*)$ associated with the index sets for the final establishment of the convergence result. As discussed in Section 5.4.1, we have redefined the index sets associated with $z \in LCP(r, M)$, where M is positive semi-definite. We can derive similar results to Lemma 5.3.4 regarding the local constancy of the index sets in the positive semi-definite case.

Lemma 5.4.7. *Let $\alpha^* = \alpha(q+\sigma_* p, M+\sigma_* N)$ and $\gamma^* = \gamma(q+\sigma_* p, M+\sigma_* N)$ be the index sets associated with z^* as defined in (5.31)–(5.32). It follows for any convergent subsequence of $\{\sigma_i\}$ satisfying $\lim_{i_k\to+\infty} \sigma_{i_k} = \sigma_*$ that if $z^* \equiv z(\sigma_*)$ is a non-degenerate solution to (5.4) with $z^*_{\alpha^*} > 0$ and $((M + \sigma_* N)z^* + (q + \sigma_* p))_{\gamma^*} > 0$, and all assumptions listed in Lemma 5.4.6 hold, then there exists an integer $K > 0$ such that z^{i_k} and $z^{i_k+0.5}$ have the same index sets as z^* for all $i > K$.*

Proof. Without loss of generality, we again denote the convergent subsequence $\{z^{i_k}\}$ as $\{z^i\}$. The result of Lemma 5.4.5 and the assumption (5.39) give

$$\begin{aligned}
&\|z^{i+0.5} - z^i\| \le \psi_i \left(|\sigma_{i+1} - \sigma_i| \left(\|p\| + \|N\|\, \|z^i\|\right)\right)^{\kappa_i},\\
&\|z^{i+1} - z^{i+0.5}\| < G|\sigma_{i+1} - \sigma^i|\|z^{i+0.5} - z^i\|.
\end{aligned}$$

Since $\{\psi_i\}$ and $\{\kappa_i\}$ are bounded, there exists $K_1 > 0$ such that for all $i > K_1$,

$$\begin{aligned}
&\|z^{i+1} - z^{i+0.5}\| \le \epsilon,\\
&\|z^{i+0.5} - z^i\| \le \epsilon,
\end{aligned}$$

for any $\epsilon > 0$ Hence, under the assumptions that

$$\begin{aligned}
&z^*_{\alpha^*} > 0,\\
&((M + \sigma_* N)z^* + (q + \sigma_* p))_{\gamma *} > 0,
\end{aligned}$$

we have there exist $K > K_1 > 0$ such that for all $i > K$,

$$\begin{aligned}
&z^i_{\alpha^*} > 0, \text{ and } \left((M + \sigma_i N)z^i + (q + \sigma_i p)\right)_{\gamma^*} > 0,\\
&z^{i+0.5}_{\alpha^*} > 0, \text{ and } \left((q - \sigma_i N z^i) + \sigma_{i+1}(p + N z^i) + (M + \sigma_i N)z^{i+0.5}\right)_{\gamma^*} > 0.
\end{aligned}$$

According to the definition of the index sets in the case that the associated matrices are positive semi-definite, the index sets of z^i and $z^{i+0.5}$ satisfy the following conditions:

$$\begin{aligned}
&\alpha^* \subseteq \alpha^i \text{ and } \gamma^* \subseteq \gamma^i, \\
&\alpha^* \subseteq \alpha^{i+0.5} \text{ and } \gamma^* \subseteq \gamma^{i+0.5},
\end{aligned}$$

for all $i > K$. Due to the non-degeneracy of z^*, the union of α^* and γ^* is the full set. Thus, we have z^i, $z^{i+0.5}$ and z^* share the same index set for all $i > K$. □

If that all assumptions listed in Lemma 5.4.7 hold, problem (5.1) is equivalent to optimization problem

$$\begin{aligned}
&\text{minimize}_{\sigma,z} \;\; f(\sigma, z) \\
&\text{such that } \; z_{\alpha^*} \geq 0 = \left((M+\sigma N)z + (q+\sigma p)\right)_{\alpha^*}, \\
&z_{\gamma^*} = 0 \leq \left((M+\sigma N)z + (q+\sigma p)\right)_{\gamma^*}, \\
&\text{and } \underline{\sigma} \leq \sigma \leq \bar{\sigma}.
\end{aligned} \tag{5.41}$$

The first-order KKT conditions of the optimization problem (5.41) can be calculated as

$$\begin{aligned}
0 = &\begin{pmatrix} \frac{\partial f(\sigma_*, z^*)}{\partial \sigma} \\ \frac{\partial f(\sigma_*, z^*)}{\partial z_1} \\ \vdots \\ \frac{\partial f(\sigma_*, z^*)}{\partial z_n} \end{pmatrix} - \theta_1^* e^1 - \sum_{j \in \alpha^*} u_j^* e^{j+1} - \sum_{j \in \gamma^*} u_j^* \begin{pmatrix} N_{j,\cdot} z + p_j \\ M_{j1} + \sigma N_{j1} \\ \vdots \\ M_{jn} + \sigma N_{jn} \end{pmatrix} \\
&+ \theta_2^* e^1 + \sum_{j \in \gamma^*} \lambda_j^* e^{j+1} + \sum_{j \in \alpha^*} \lambda_j^* \begin{pmatrix} N_{j,\cdot} z + p_j \\ M_{j1} + \sigma N_{j1} \\ \vdots \\ M_{jn} + \sigma N_{jn} \end{pmatrix},
\end{aligned} \tag{5.42}$$

$$\begin{aligned}
&0 \leq u_{\alpha^*} \perp z_{\alpha^*} \geq 0, \\
&0 \leq u_{\gamma^*} \perp \left((M+\sigma N)z + (q+\sigma p)\right)_{\gamma^*} \geq 0, \\
&0 \leq \theta_1^* \perp \sigma - \underline{\sigma} \geq 0, \\
&0 \leq \theta_2^* \perp \bar{\sigma} - \sigma \geq 0.
\end{aligned} \tag{5.43}$$

Here, e^j is the j-th coordinate vector of size $n+1$ and $N_{j,\cdot}$ is the j-th row of the matrix N. Under mild assumptions, we can show that the pair $(\sigma_*, z(\sigma_*))$ satisfies the first-oder KKT conditions (5.42)–(5.43) for any interior limit point σ_* of the sequence $\{\sigma_i\}$.

Theorem 5.4.8. *For any convergent subsequence of* $\{\sigma_i\}$ *satisfying* $\lim_{i_k \to +\infty}\{\sigma_{i_k}\} = \sigma_*$, *let* $z^* \equiv z(\sigma_*)$. *If (i) the assumptions listed in Lemma 5.4.7 hold; (ii) the derivative of* f *is a Lipschitz function with Lipschitz constant* L*; (iii) the limit point* σ_* *is an interior point of the feasible range* $\Gamma = [\underline{\sigma}, \bar{\sigma}]$*; and (iv)* $\{e^{j+1}\}$, $j \in \gamma^*$, *and*

$$\begin{pmatrix} N_{j,\cdot}z^* + p_j \\ M_{j1} + \sigma_* N_{j1} \\ \vdots \\ M_{jn} + \sigma_* N_{jn} \end{pmatrix},$$

$j \in \alpha^*$, *are independent vectors, then* (σ_*, z^*) *satisfies the first-order KKT conditions of the optimization problem (5.41).*

Proof. Without loss of generality, we use $\{\sigma_i\}$ to denote the convergent subsequence $\{\sigma_{i_k}\}$. Using the result of Lemma 5.4.7, we have that there exists $K_1 > 0$ such that $z^{i+0.5}$ shares the same index sets as z^* for all $i > K_1$. Therefore, $(\sigma_{i+1}, z^{i+0.5})$ is the optimal solution to optimization problem

$$\begin{aligned} &\text{minimize}_{\sigma,z} \;\; f(\sigma, z) + \frac{c}{2}[(\sigma - \sigma_i)^2 + \|z - z^i\|^2] \\ &\text{such that } \; z_{\alpha^*} \geq 0 = ((q - \sigma_i N z^i) + \sigma(p + Nz^i) + (M + \sigma_i N)z)_{\alpha^*}, \\ &\; z_{\gamma^*} = 0 \leq ((q - \sigma_i N z^i) + \sigma(p + Nz^i) + (M + \sigma_i N)z)_{\gamma^*}, \\ &\text{and } \underline{\sigma} \leq \sigma \leq \bar{\sigma}. \end{aligned} \tag{5.44}$$

Hence, $(\sigma_{i+1}, z^{i+0.5})$ satisfies the following KKT conditions (5.45)–(5.46) of problem (5.44):

$$\begin{aligned} 0 = \begin{pmatrix} \frac{\partial f(\sigma_{i+1}, z^{i+0.5})}{\partial \sigma} + c(\sigma_{i+1} - \sigma_i) \\ \frac{\partial f(\sigma_{i+1}, z^{i+0.5})}{\partial z_1} + c(z_1^{i+0.5} - z_1^i) \\ \vdots \\ \frac{\partial f(\sigma_{i+1}, z^{i+0.5})}{\partial z_n} + c(z_n^{i+0.5} - z_n^i) \end{pmatrix} &- \theta_1^i e^1 - \sum_{j\in\alpha^*} u_j^i e^{j+1} - \sum_{j\in\gamma^*} u_j^i \begin{pmatrix} N_{j,\cdot}z^i + p_j \\ M_{j1} + \sigma_i N_{j1} \\ \vdots \\ M_{jn} + \sigma_i N_{jn} \end{pmatrix} \\ &+ \theta_2^i e^1 + \sum_{j\in\gamma^*} \lambda_j^i e^{j+1} + \sum_{j\in\alpha^*} \lambda_j^i \begin{pmatrix} N_{j,\cdot}z^i + p_j \\ M_{j1} + \sigma_i N_{j1} \\ \vdots \\ M_{jn} + \sigma_i N_{jn} \end{pmatrix}, \end{aligned} \tag{5.45}$$

$$
\begin{aligned}
&0 \le u^i_{\alpha^*} \perp z^{i+0.5}_{\alpha^*} \ge 0, \\
&0 \le u^i_{\gamma^*} \perp ((q - \sigma_i N z^i) + \sigma_{i+1}(p + N z^i) + (M + \sigma_i N) z^{i+0.5})_{\gamma^*} \ge 0, \\
&0 \le \theta^i_1 \perp \sigma_{i+1} - \underline{\sigma} \ge 0, \\
&0 \le \theta^i_2 \perp \bar{\sigma} - \sigma_{i+1} \ge 0.
\end{aligned}
\tag{5.46}
$$

for any $i > K_1$. Notice that $z^*_{\alpha^*} > 0$ and $((M + \sigma_* N) z^* + (q + \sigma_* p))_{\gamma *} > 0$. Besides, $z^{i+0.5} \to z^*$ and $z^i \to z^*$ as $i \to +\infty$. Hence, there exists $K_2 > K_1$ such that $\left((q - \sigma_i N z^i) + \sigma_{i+1}(p + N z^i) + (M + \sigma_i N) z^{i+0.5}\right)_{\gamma^*} > 0$ and $z^{i+0.5}_{\alpha^*} > 0$ for all $i > K_2$. Then the complementarity constraints in the KKT conditions (5.46) and the assumption that σ_* is an interior point imply

$$\theta^i_{1,2} = 0, \text{ and } u^i_j = 0,$$

for all $i > K_2$ and $j \in \{1, 2, \ldots, n\}$.

Since the derivative of f is Lipschitz continuous, the following holds for all i:

$$
\begin{aligned}
&\| \bigtriangledown f(\sigma_*, z^*) - \bigtriangledown f(\sigma_{i+1}, z^{i+0.5}) \| \\
&= \| \bigtriangledown f(\sigma_*, z^*) - \bigtriangledown f(\sigma_*, z^{i+0.5}) + \bigtriangledown f(\sigma_*, z^{i+0.5}) - \bigtriangledown f(\sigma_{i+1}, z^{i+0.5}) \| \\
&\le L \left(\|z^* - z^{i+0.5}\| + |\sigma_* - \sigma_{i+1}| \right).
\end{aligned}
$$

Therefore, we have $\bigtriangledown f(\sigma_{i+1}, z^{i+0.5}) \to \bigtriangledown f(\sigma_*, z^*)$ as $i \to +\infty$.

Furthermore, under the assumption that $\{e^{j+1}\}$, $j \in \gamma^*$, and

$$
\begin{pmatrix}
N_{j,\cdot} z^* + p_j \\
M_{j1} + \sigma_* N_{j1} \\
\vdots \\
M_{jn} + \sigma_* N_{jn}
\end{pmatrix},
$$

$j \in \alpha^*$ are independent, we can show the boundedness of $\{\lambda^i\}$. Similar to the proof of Lemma 5.4.4, we use the contradiction to prove our claim. Assume $\{\lambda^i\}$ is not bounded, we define $\{\widehat{\lambda^i}\}$ by

$$\widehat{\lambda^i_j} \equiv \frac{\lambda^i_j}{\sum_j |\lambda^j_i|}, \ \forall j \in \{1, 2, \ldots, n\}.$$

Therefore, we have $\sum_j |\widehat{\lambda^i_j}| = 1$ and $\{\widehat{\lambda^i}\}$ is bounded. We select a convergent subsequence $\{\widehat{\lambda^{i_l}}\}$ of $\{\widehat{\lambda^i}\}$.

Dividing both sides of equation (5.45) by $\frac{1}{\sum_j |\lambda_j^{i_l}|}$ and letting $i_l \to +\infty$ lead to

$$0 = \sum_{j\in\gamma^*} \widehat{\lambda_j^*} e^{j+1} + \sum_{j\in\alpha^*} \widehat{\lambda_j^*} \begin{pmatrix} N_{j,\cdot} z^* + p_j \\ M_{j1} + \sigma_* N_{j1} \\ \vdots \\ M_{jn} + \sigma_* N_{jn} \end{pmatrix}. \tag{5.47}$$

Here,

$$\widehat{\lambda_j^*} = \lim_{i_l \to +\infty} \widehat{\lambda_j^{i_l}}, \ \ \forall j \in \{1, 2, \dots, n\} \text{ and } \sum_j |\widehat{\lambda_j^*}| = 1.$$

Since $\widehat{\lambda^*} \neq 0$, condition (5.47) contradicts with the assumption that vectors $\{e^{j+1}\}$ and

$$\begin{pmatrix} N_{j,\cdot} z^* + p_j \\ M_{j1} + \sigma_* N_{j1} \\ \vdots \\ M_{jn} + \sigma_* N_{jn} \end{pmatrix}$$

are independent. Hence, $\{\lambda^i\}$ is bounded. Let $\{\lambda^{i_k}\}$ be a convergent subsequence of $\{\lambda^i\}$ with

$$\lim_{i_k \to +\infty} \lambda^{i_k} = \lambda^*. \tag{5.48}$$

As $i_k \to +\infty$, the following conditions hold:

$$\begin{aligned} &(i) \nabla f(\sigma_{i_k+1}, z^{i_k+0.5}) \to \nabla f(\sigma_*, z^*); \\ &(ii) z^{i_k+0.5} - z^{i_k} \to 0, \text{ and } \sigma_{i_k+1} - \sigma_{i_k} \to 0; \\ &(iii) \theta_{1,2}^{i_k} = 0, \ u_j^{i_k} = 0, \ \forall j \in \{1, 2, \dots, n\}; \\ &(iv) \lim_{i_k \to +\infty} \lambda^{i_k} = \lambda^*. \end{aligned}$$

Combining (i)–(iii) with the KKT conditions (5.42)–(5.43) together, we have (σ_*, z^*) satisfies conditions

$$0 = \begin{pmatrix} \frac{\partial f(\sigma_*, z^*)}{\partial \sigma} \\ \frac{\partial f(\sigma_*, z^*)}{\partial z_1} \\ \vdots \\ \frac{\partial f(\sigma_*, z^*)}{\partial z_n} \end{pmatrix} - \theta_1^* e^1 - \sum_{j \in \alpha^*} u_j^* e^{j+1} - \sum_{j \in \gamma^*} u_j^* \begin{pmatrix} N_{j,\cdot} z^* + p_j \\ M_{j1} + \sigma_* N_{j1} \\ \vdots \\ M_{jn} + \sigma_* N_{jn} \end{pmatrix}$$

$$+ \theta_2^* e^1 + \sum_{j \in \gamma^*} \lambda_j^* e^{j+1} + \sum_{j \in \alpha^*} \lambda_j^* \begin{pmatrix} N_{j,\cdot} z^* + p_j \\ M_{j1} + \sigma_* N_{j1} \\ \vdots \\ M_{jn} + \sigma_* N_{jn} \end{pmatrix},$$

$$\begin{aligned} & 0 \le u_{\alpha^*} \perp z_{\alpha^*} \ge 0, \\ & 0 \le u_{\gamma^*} \perp ((M + \sigma N) z + (q + \sigma p))_{\gamma^*} \ge 0, \\ & 0 \le \theta_1^* \perp \sigma - \underline{\sigma} \ge 0, \\ & 0 \le \theta_2^* \perp \bar{\sigma} - \sigma \ge 0, \end{aligned}$$

with

$$\theta_{1,2}^* = 0, \text{ and } u_j^* = 0, \ \forall j \in \{1, 2, \ldots, n\}.$$

□

The proof of the KKT conditions satisfied by a limit point relies on the local constancy of the index sets associated with a limit point and the independence of $\{e^{j+1}\}$, $j \in \gamma^*$ and $\begin{pmatrix} N_{j,\cdot} z^* + p_j \\ M_{j1} + \sigma_* N_{j1} \\ \vdots \\ M_{jn} + \sigma_* N_{jn} \end{pmatrix}$, $j \in \alpha^*$. If the limit point σ_* of a convergent subsequence $\{\sigma_{i_k}\}$ is a non-degenerate interior point of Γ, then z^{i_k}, $z^{i_k+0.5}$ share the same index sets with z^* when i_k is sufficiently large. Therefore, the KKT conditions satisfied by $z^{i_k+0.5}$ and z^* share similar forms. Furthermore, the independence of $\{e^{j+1}\}$ and $\begin{pmatrix} N_{j,\cdot} z^* + p_j \\ M_{j1} + \sigma_* N_{j1} \\ \vdots \\ M_{jn} + \sigma_* N_{jn} \end{pmatrix}$ guarantees the boundedness of the Lagrange multipliers $\{\lambda^{i_k}\}$. Hence, the KKT conditions of the limit point come naturally by taking the limit of the KKT conditions of $\{(\sigma_{i_k}, z^{i_k+0.5})\}$.

5.5 Numerical experiments

In this section, we evaluate the numerical performance of the newly proposed Algorithm 10 and Algorithm 12 along with a one-dimensional grid search method (Algorithm 9) and IMPA (Algorithm 6)for solving doubly uni-parametric MPCCs (5.1) where the associated matrices are positive definite and positive semi-definite. In our implementation, all four methods are coded in MATLAB. Throughout our numerical experiments, matrices M and N are tridiagonal matrices, p and q are randomly generated vectors. The sizes of our problems vary among

$$40,\ 120,\ 360, 1080.$$

In terms of the feasible range, we fix $\underline{\sigma}$ at 0 and let $\bar{\sigma}$ be 0.5, 1, 2, and 8, respectively. Also, the objective functions in all tests take the same simple form

$$f(\sigma, z) = (z - z_{\text{obs}})^T (z - z_{\text{obs}}),$$

where z_{obs} is pre-calculated by adding the solution of the $LCP(q + \sigma p, M + \sigma N)$ at $\sigma = 2$ with a small perturbation. Given the objective function $f(\sigma, z)$, our goal in this set of experiments is to investigate whether our proposed Algorithm 10, Algorithm 12, the grid search method and IMPA are able to obtain an objective value that is close to zero. Ideally, we would also like the algorithms to reproduce the result $\sigma = 2$ ideally. However, since the constrained problem (5.1) might not have a unique optimal solution, we are satisfied if the algorithms produce small objective values.

Before presenting the computational results, we provide some implementation details of the above four algorithms. In Algorithm 10, the major computation at each iteration involves two steps. At the first step, we recover the feasibility of (σ^{i+1}, z^{i+1}) by solving the $LCP(q + \sigma_{i+1} p, M + \sigma_{i+1} N)$. In our tests, we use the PSOR method to solve the corresponding LCPs. Similar to the implementation of the PSOR method discussed in Section 3.5, we set the stopping criterion of the PSOR method as

$$\| \min \left(z^{i+1}, q + \sigma_{i+1} p + (M + \sigma_{i+1} N) z^{i+1} \right) \|_\infty \leq \epsilon_{\text{sor}}.$$

Here, ϵ_{sor} is the error tolerance level of the PSOR method and set as 10^{-6} in our implementation. The ω parameter of the PSOR method is fixed at 1.5 throughout our experiments. The second step of Algorithm 10 involves the search of a direction which could potentially lead to the decrease of the objective function. At each iteration i, we use the pair (σ_i, z^i) to construct a sub-problem of form (5.6). We then solve the sub-problem (5.6) by breaking the feasible region $\Gamma = [\underline{\sigma}, \bar{\sigma}]$ into smaller sub-intervals and computing the sub-problem as a quadratic program in each sub-interval. The validity of the transformation is simply based on the local constancy of the index sets. Each one-dimensional quadratic program is then be solved in MATLAB by comparing the objective values at two or three points, two end points of the feasible range and one point which is the solution to $\bigtriangledown f(\sigma) = 0$ (if it is in the feasible range). In our implementation, we adopt a pivoting method (Algorithm 11) to determine the partition of the feasible range. We again use the PSOR method to solve the $LCP\left((q - \sigma_i N z^i) + \underline{\sigma}(p + N z^i), M + \sigma_i N\right)$, and the Gaussian elimination method to calculate the matrix inverse in Algorithm 11.

At each iteration of Algorithm 12, the main computation is similar to that of Algorithm 10. However, there are a few differences. (i) In the direction search step of Algorithm 12, other than the partition of the feasible range and the computation of a quadratic program in each sub-interval, we need to solve one extra quadratic program at each break point. This modification is mainly due to the fact that LCP (5.35) at each break point does not necessarily have a unique solution. The additional calculation avoids the possibility of missing optimal solutions at break points. (ii) For the same reason, we modify the recovery step of Algorithm 12 by selecting $z \in SOL(q + \sigma_{i+1} p, M + \sigma_{i+1} N)$ which is closest to $z^{i+0.5}$ as the new starting point of the next iteration.

We impose the following stopping criterion on both Algorithm 10 and Algorithm 12. The algorithms are terminated when either one of the following conditions is satisfied:

$$\begin{aligned} & i > N_{\max}, \\ & |f(\sigma_i, z^i)| \leq \epsilon_{\text{linear}}, \\ & |\sigma_i - \sigma_{i-1}| \leq \epsilon_{\text{linear}}. \end{aligned}$$

Notice that we impose a maximum number of iterations $N_{\max}$ to both algorithms in the above stopping criterion. If none σ_i satisfies the other two termination conditions after $N_{\max}$ iterations, we then use σ_i which leads to the smallest objective value so far as the final result. Also, both algorithms share the same error tolerance level ϵ_{linear}. Throughout our numerical experiments, we set $N_{\max} = 40$ and $\epsilon_{\text{linear}} = 10^{-6}$. The choice of the above two parameters yields satisfactory numerical results.

As for the implementation of IMPA, similar to the implementation of IMPA discussed in Section 4.4, we let W be an identity matrix for all iterations. In particular, W equals one in our problem since the dimension of σ equals one. The termination test for IMPA is given as

$$\min\left(\| \, d\sigma \, \|, f(\sigma)\right) \leq \epsilon_{\text{impa}}.$$

Here, ϵ_{impa} is the error tolerance level of IMPA and we set it the same as the tolerance level of Algorithm 12 and Algorithm 13.

$$\epsilon_{\text{impa}} = \epsilon_{\text{linear}} = 10^{-6}.$$

We also impose a maximum number of 2000 iterations on IMPA. In the direction generation step of IMPA, we again use the Gaussian elimination method to calculate the expression of dz in terms of $d\sigma$. Once we write dz in terms of $d\sigma$, we simplify the convex quadratic program used for the direction generation into a quadratic program and solve the quadratic program in MATLAB by comparing the objective function at two or three points. Recall that to determine the step size at each iteration ν, the evaluation of the objective function f at a sequence of points around σ^ν is required in IMPA. Therefore, we evaluate the objective function by solving a sequence of LCPs via the PSOR method, whose setting is the same as those used in Algorithm 10 and Algorithm 12.

In terms of the implementation of the one dimension grid search method, a total number of $m + 1$ LCPs need to be solved for a given discretization of the feasible range Γ: $(\sigma_0, \ldots, \sigma_m)$. We again use the PSOR method to solve the corresponding LCPs for all end points of the sub-intervals. The application of the same LCP solver in the four methods ensures a fair numerical comparison.

5.5.1 Algorithm comparison for positive definite matrices M and N

In this section, we report the numerical performance of three methods: Algorithm 10, a one-dimensional grid search method and IMPA for solving doubly uni-parametric MPCCs when the associated matrices M and N are positive definite. We begin with an introduction to the construction of the MPCCs, followed by the numerical results. As in [FLMN10], we use the matrices derived under the BSM model via the finite element method to generate the matrices M and N. Specifically, M and N are tridiagonal matrices of the following form:

$$M \equiv \frac{h}{6} \begin{pmatrix} 4 & 1 & \ldots & 0 \\ 1 & 4 & \ldots & \vdots \\ \vdots & \vdots & \ddots & 1 \\ 0 & \ldots & 1 & 4 \end{pmatrix}, \quad N \equiv \begin{pmatrix} a_0 & a_1 & \ldots & 0 \\ a_{-1} & a_0 & \ldots & \vdots \\ \vdots & \vdots & \ddots & a_1 \\ 0 & \ldots & a_{-1} & a_0 \end{pmatrix},$$

where the entries of the matrix N are

$$\begin{aligned} a_0 &= dt(\frac{2}{3}rh + \frac{1}{h}\zeta^2), \\ a_{\pm 1} &= \mp\frac{1}{2}\mu + \frac{1}{6}rh - \frac{1}{2h}\zeta^2, \end{aligned}$$

with $\mu = r - q - \frac{1}{2}\zeta^2$. In our experiment, we set the parameters as

$$r = 0.4,\ q = 0.0,\ \zeta = 0.3,\ h = 0.045,\ dt = 0.01,$$

for the generation of the matrices M and N. We also let the sizes of the matrices M and N vary from 120, 360 to 1080. The vectors q and p in the MPCC (5.1) are generated randomly, each having the same size as the matrices M and N. We set the parameter c used in the objective function to 0, although this choice could presumably affect the practical performance of Algorithm 10. In our implementation, this choice yields satisfactory results with a monotonically decreasing objective value. We test Algorithm 10, the one-dimensional grid search method and IMPA for solving the doubly uni-parametric MPCC constructed above in MATLAB with feasible ranges $[\underline{\sigma}, \bar{\sigma}]$ defined for four different values of $\bar{\sigma}$. For all the numerical results, we report the computational time in seconds (CPU), the calculated optimal σ (Cal opt) and the corresponding value (Obj val) of the objective function of each algorithm. We also include the number of LCPs solved by each algorithm in our report for a thorough comparison of the three methods. Details are summarized in Tables 5.1–5.3.

We note that Algorithm 10 is quite efficient in finding satisfactory results as suggested by Tables 5.1–5.3. In most of the examples, Algorithm 10 yields smaller objective values compared to both the grid search method and IMPA for solving doubly uni-parametric MPCCs (5.1) of different sizes and feasible ranges. Furthermore, in most of the test examples, the computational time required by Algorithm 10 to calculate a solution to the MPCC is less than or on the same level as the computational time required by IMPA, both of which require significantly shorter time than needed by the grid search method.

5.5.2 Algorithm comparison for positive semi-definite matrices M and N

In this section, we report the numerical performance of three methods: Algorithm 12, a one-dimensional grid search method and IMPA for solving doubly uni-parametric MPCCs when the associated matrices M and N are positive semi-definite. In our experiments, we only consider a special subclass of positive semi-definite

		Algo 10	Grid	IMPA
$\bar{\sigma} = 0.5$	Cal opt	0.5000	0.5000	0.4400
	CPU(s)	0.0392	0.3597	0.2695
	Obj val	1.87E7	1.87E7	2.14E7
	LCP	2	1001	425
$\bar{\sigma} = 1$	Cal opt	1.000	1.000	0.9400
	CPU(s)	0.0800	0.6553	0.3229
	Obj val	5.91E6	5.91E6	6.88E6
	LCP	30	2001	563
$\bar{\sigma} = 2$	Cal opt	2.000	2.000	2.000
	CPU(s)	0.1136	1.2994	0.4175
	Obj val	0.0060	0.0060	0.0060
	LCP	67	4001	702
$\bar{\sigma} = 8$	Cal opt	2.000	1.999	2.000
	CPU(s)	0.1889	8.7721	0.2303
	Obj val	0.0060	0.1089	0.0060
	LCP	178	16001	325

Table 5.1: Tridiagonal positive definite matrix: size= 120.

		Algo 10	Grid	IMPA
$\bar{\sigma} = 0.5$	Cal opt	0.0000	0.5000	0.3017
	CPU(s)	0.0838	1.0577	0.3109
	Obj val	1.833E8	5.62E7	8.76E7
	LCP	7	1001	187
$\bar{\sigma} = 1$	Cal opt	0.7093	1.000	0.8017
	CPU(s)	0.7659	1.8713	0.7307
	Obj val	3.61E7	1.77E7	2.93E7
	LCP	196	2001	325
$\bar{\sigma} = 2$	Cal opt	2.000	2.000	1.801
	CPU(s)	0.7156	3.7278	0.8312
	Obj val	0.1795	0.1795	4.20E5
	LCP	187	4001	463
$\bar{\sigma} = 8$	Cal opt	2.000	1.999	2.000
	CPU(s)	1.8931	27.2228	0.6295
	Obj val	0.1975	0.1975	0.1975
	LCP	520	16001	325

Table 5.2: Tridiagonal positive definite matrix: size= 360.

		Algo 10	Grid	IMPA
$\bar{\sigma} = 0.5$	Cal opt	0.5000	0.5000	0.4205
	CPU(s)	0.0963	6.4691	2.6289
	Obj val	1.6861E8	1.6861E8	2.0088E8
	LCP	3	1001	369
$\bar{\sigma} = 1$	Cal opt	1.000	1.000	0.9910
	CPU(s)	5.3199	11.9164	7.9719
	Obj val	5.3159E7	5.3159E7	5.3578E7
	LCP	196	2001	1161
$\bar{\sigma} = 2$	Cal opt	1.9980	2.000	1.9970
	CPU(s)	10.6526	23.4529	10.9548
	Obj val	115.3740	0.5384	252.1154
	LCP	15.8192	4001	1297
$\bar{\sigma} = 8$	Cal opt	1.9980	1.9990	2.000
	CPU(s)	15.1868	158.3096	3.1297
	Obj val	115.3746	0.9799	0.5384
	LCP	561	16001	325

Table 5.3: Tridiagonal positive definite matrix: size= 1080.

matrices: asymmetric tridiagonal positive semi-definite matrices M and N of form

$$H = \begin{pmatrix} a_0 & a_1 & 0 & \dots & 0 \\ -a_1 & a_0 & a_1 & \ddots & \vdots \\ 0 & -a_1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & a_0 & a_1 \\ 0 & \dots & 0 & -a_1 & 0 \end{pmatrix},$$

with $a_0 > 0$. It is easy to verify that any matrix H of the above form is a positive semi-definite matrix. We know that any square matrix can be written as the sum of a symmetric matrix and a skew symmetric matrix. Here, a matrix A is skew symmetric if and only if

$$-A = A^T.$$

Let H_{sym} be the symmetric matrix of H and H_{skew} be the skew symmetric matrix of H. Then H_{sym} and H_{skew} can be calculated by

$$H_{\text{sym}} = \frac{1}{2}\left(H + H^T\right),$$
$$H_{\text{skew}} = \frac{1}{2}\left(H - H^T\right).$$

		Algo 12	Grid	IMPA
$\bar{\sigma} = 0.5$	Cal opt	0.5000	0.5000	0.5000
	CPU(s)	0.9498	1.4003	1.2491
	Obj val	2.27E3	2.27E3	2.27E3
	LCP	6	1001	2615
$\bar{\sigma} = 1$	Cal opt	1.0000	1.0000	1.0000
	CPU(s)	1.1833	2.5936	1.5620
	Obj val	7.59E3	7.59E3	7.59E3
	LCP	7	2001	2753
$\bar{\sigma} = 2$	calculated opt	2.000	2.000	1.9999
	CPU(s)	1.0574	4.9217	1.4379
	Obj val	7.87E-11	7.89E-11	9.93E-7
	LCP	18	4001	2129
$\bar{\sigma} = 8$	calculated opt	2.000	1.9998	2.0023
	CPU(s)	1.5100	18.4792	0.1043
	Obj val	1.69E-11	7.30E-6	0.03
	LCP	74	16001	63

Table 5.4: Tridiagonal positive semi-definite matrix: size= 40.

Moreover, the following condition holds for any real-valued vector x:

$$x^T H_{\text{skew}} x = x^T H_{\text{skew}}^T x = -x^T H_{\text{skew}} x.$$

Therefore, we have $x^T H_{\text{skew}} x = 0$ and $x^T H x = x^T H_{\text{sym}} x + x^T H_{\text{skew}} x = x^T H_{\text{sym}} x$. Due to the construction of the matrix H, we have $x^T H_{\text{sym}} x \geq 0$ for any x. Hence, H is a positive semi-definite matrix.

In our experiment, the diagonal entry a_0 of the matrix M is set as 2, and the off-diagonal entry a_1 is set as 0.6. The diagonal entry of the matrix N is set as 0.8, and the off-diagonal entry a_1 is set as -0.3. We let the sizes of the matrices M and N vary from 40, 120 to 360. The vectors q and p in MPCC (5.1) are two randomly generated vectors, each having the same size as the matrices M and N. We again set the parameter c used in the objective function to 0, which yields satisfactory results with a monotonically decreasing objective value in our implementation. We test Algorithm 12, the one-dimensional grid search method and IMPA for solving the doubly uni-parametric MPCC constructed above in MATLAB with feasible ranges $[\underline{\sigma}, \bar{\sigma}]$ defined for four different values of $\bar{\sigma}$. Similar to the numerical report in Section 5.5.2, we report the computational time in seconds (CPU), the calculated optimal σ (Cal opt) and the corresponding value (Obj val) of the objective function of each algorithm for all numerical tests. We also include the number of LCPs solved by each algorithm in our report for a thorough comparison of the three methods. Details are summarized in Tables 5.4–5.6.

Observing the numerical results of Tables 5.4–5.6, we notice that they are similar to the numerical results

		Algo 12	Grid	IMPA
$\bar{\sigma} = 0.5$	Cal opt	0.5000	0.5000	0.5000
	CPU(s)	0.9333	1.7900	2.0435
	Obj val	6.63E3	6.63E3	6.63E3
	LCP	6	1001	3073
$\bar{\sigma} = 1$	Cal opt	1.0000	1.0000	1.0000
	CPU(s)	1.0213	3.6845	2.2066
	Obj val	2.21E3	2.21E3	2.21E3
	LCP	6	2001	3211
$\bar{\sigma} = 2$	Cal opt	1.9999	1.9999	1.9999
	CPU(s)	1.3255	6.8582	1.9877
	Obj val	1.43E-12	1.43E-12	1.43E-12
	LCP	13	4001	2465
$\bar{\sigma} = 8$	Cal opt	2.0000	1.9998	2.0001
	CPU(s)	2.4692	27.8158	0.1783
	Obj val	1.47E-7	2.12E-5	3.85E-4
	LCP	1028	16001	95

Table 5.5: Tridiagonal positive semi-definite matrix: size= 120.

		Algo 12	Grid	IMPA
$\bar{\sigma} = 0.5$	Cal opt	0.5000	0.5000	0.5000
	CPU(s)	1.1834	3.1949	4.4436
	Obj val	1.99E4	1.99E4	1.99E4
	LCP	6	1001	2615
$\bar{\sigma} = 1$	Cal opt	1.0000	1.0000	1.0000
	CPU(s)	1.0156	5.7687	4.8586
	Obj val	6.65E3	6.65E3	6.65E3
	LCP	6	2001	2753
$\bar{\sigma} = 2$	Cal opt	2.0000	2.0000	2.0000
	CPU(s)	1.42230	11.3678	3.8823
	Obj val	4.43E-7	4.43E-7	4.43E-7
	LCP	13	4001	2347
$\bar{\sigma} = 8$	Cal opt	1.9999	1.9998	2.000
	CPU(s)	12.0796	25.8007	1.0290
	Obj val	4.43E-7	9.62E-5	03.93E-8
	LCP	340	16001	351

Table 5.6: Tridiagonal positive semi-definite matrix: size= 360.

obtained in Section 5.5.2. In most of the examples, Algorithm 12 yields smaller objective values compared to both the grid search method and IMPA for solving doubly uni-parametric MPCCs (5.1) of different sizes and feasible ranges. Furthermore, in most of the test examples, the computational time required by Algorithm 12 to calculate a solution to the MPCC is less than or on the same level as the computational time required by IMPA, both of which require significantly shorter time than needed by the grid search method. One special case where IMPA outperforms Algorithm 12 is when $\bar{\sigma} = 8$, IMPA uses less time than Algorithm 12 to find solutions with similar accuracies.

Chapter 6

Conclusion

In this dissertation, we first discuss the numerical evaluation of American options under three models, a local volatility model and two jump diffusion models: Kou's jump diffusion model and the Dupire system. The jump diffusion models capture the fluctuation of the underlying assets by introducing a Levy process as the jump part, and the local volatility model relaxes the constant volatility assumption of the Black-Scholes-Merton (BSM) model. In Chapter 2, we establish partial differential complementarity systems for pricing American options under the three aforementioned models, followed by the introduction of two different discretization schemes, a finite difference method and a finite element method. We utilize the two different discretization schemes to transform the complementarity systems into a collection of linear complementarity problems (LCPs). Once we reformulated the pricing problems of American options as a collection of LCPs, we apply four existing numerical algorithms—a PSOR method, a two phase active-set method, a semi-smooth Newton method and a parametric pivoting method—to find approximate solutions of the LCPs. Details of the algorithms are discussed in Chapter 3 along with the numerical testing results of the four methods for solving LCPs derived under Kou's jump diffusion model and the Dupire system. The numerical results further suggest that the two phase active-set method outperforms the other three methods by solving the corresponding LCPs most efficiently throughout all our numerical examples.

In this thesis, we also study state-of-the-art methods and computational techniques for calibrating American options under the models discussed in Chapter 2. In particular, we calibrate the volatility parameter of American options under the Dupire system and the local volatility model via two approaches, an implicit programming algorithm (IMPA) and a new hybrid algorithm in Chapter 4. Both approaches are developed based on the simple observation that the calibration problems of American options are the inverse problems of the forward pricing problems. Hence, based on the formulation of the forward pricing problems as LCPs, we can model our calibration problems as mathematical programs with complementarity constraints (MPCCs). We provide detailed description of both IMPA and the hybrid algorithm and the numerical performance of the two methods for solving MPCCs derived under the Dupire system and the local volatility model. Specifically, we compare the numerical results of IMPA with a one-dimensional grid search method for solving MPCCs derived under the Dupire system. The results of our experiments suggest that IMPA performs as good as the grid search method for solving this type of MPCCs. As for solving the MPCCs derived under the local volatility model, where the dimension of the parameter is too large for the direct use of a grid search method, IMPA still gives considerably good numerical performance. Especially with the synthetic data, the final objective value generated by IMPA is very small. We also test the new hybrid algorithm, which combines the grid search method together with IMPA, for solving MPCCs derived under the local volatility model with the same set of synthetic data used in the tests of IMPA. The new algorithm yields satisfactory results in our experiments and demonstrates its capability of accelerating the convergence for solving MPCCs of large sizes.

Inspired by the calibration problems of American options under the BSM model, we study a special subclass of MPCCs: a doubly uni-parametric MPCC (5.1) in Chapter 5. We present two novel approaches, Algorithm 10 and Algorithm 12 to solve this type of MPCCs in this work. Specifically, we use Algorithm 10 to solve MPCCs (5.1) when the associated matrices M and N are positive definite, and Algorithm 12 to solve MPCCs (5.1) when the associated matrices M and N are merely positive semi-definite. Both new algorithms are designed by searching descent directions via solving a sequence of sub-problems, whose constraints are linear approximations to the doubly uni-parametric complementarity constraints in MPCC (5.1). We test both new algorithms along with a grid search method and IMPA for solving MPCCs (5.1) with tridiagonal positive definite and positive semi-definite matrices in our experiments. The numerical reports suggest that both new algorithms are able to produce satisfactory results efficiently. In the latter part of Chapter 5, we further develop the convergence results for the two new algorithms by studying the special structures of the constraints in the MPCC (5.1) . We show that under mild conditions, any non-degenerate limit point is a B-stationarity point of the objective function generated by Algorithm 10 and satisfies the first-order KKT conditions of the objective function generated by Algorithm 12. The essence of the convergence analysis lies in the local stability of the limit point as a function of the uni-parameter.

References

[A$^+$66] Larry Armijo et al. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3, 1966.

[AA00] Leif Andersen and Jesper Andreasen. Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing. *Review of Derivatives Research*, 4(3):231–262, 2000.

[Ach08] Yves Achdou. An inverse problem for a parabolic variational inequality with an integro-differential operator. *SIAM Journal on Control and Optimization*, 47(2):733–767, 2008.

[AIP$^+$04] Yves Achdou, Govindaraj Indragoby, Olivier Pironneau, et al. Volatility calibration with american options. *Methods and Applications of Analysis*, 11(4):533–556, 2004.

[AP05] Yves Achdou and Olivier Pironneau. *Computational methods for option pricing*, volume 30. Siam, 2005.

[Ave98] Marco Avellaneda. Minimum-relative-entropy calibration of asset-pricing models. *International Journal of theoretical and applied finance*, 1(04):447–472, 1998.

[Bat96] David S Bates. Jumps and stochastic volatility: Exchange rate processes implicit in deutsche mark options. *Review of financial studies*, 9(1):69–107, 1996.

[Bec81] Stan Beckers. Standard deviations implied in option prices as predictors of future stock price variability. *Journal of Banking & Finance*, 5(3):363–381, 1981.

[BL05] Artan Borici and Hans-Jakob Lüthi. Fast solutions of complementarity formulations in American put pricing. *Journal of Computational Finance*, 9:63–81, 2005.

[BS73] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *The journal of political economy*, pages 637–654, 1973.

[CFH93] Robert Cumby, Stephen Figlewski, and Joel Hasbrouck. Forecasting volatilities and correlations with egarch models. *The Journal of Derivatives*, 1(2):51–63, 1993.

[CLV99] Thomas F Coleman, Yuying Li, and Arun Verma. Reconstructing the unknown volatility function. *Journal of Computational Finance*, 2(3):77–102, 1999.

[CPS09] Richard W Cottle, Jong-Shi Pang, and Richard E Stone. *The linear complementarity problem*. SIAM, 2009.

[DK94] Emanuel Derman and Iraj Kani. Riding on a smile. *Risk*, pages 32–39, February 1994.

[DLFK96] Tecla De Luca, Francisco Facchinei, and Christian Kanzow. A semismooth equation approach to the solution of nonlinear complementarity problems. *Mathematical Programming*, 75(3):407–439, 1996.

[DPS00] Darrell Duffie, Jun Pan, and Kenneth Singleton. Transform analysis and asset pricing for affine jump-diffusions. *Econometrica*, 68(6):1343–1376, 2000.

[Dup94] Bruno Dupire. Pricing with a smile. *Risk*, pages 18–20, January 1994.

[EG02] Louis H Ederington and Wei Guan. Why are those options smiling? *The Journal of Derivatives*, 10(2):9–34, 2002.

[Fis92] Andreas Fischer. A special newton-type optimization method. *Optimization*, 24(3-4):269–284, 1992.

[FJQ99] Francisco Facchinei, Houyuan Jiang, and Liqun Qi. A smoothing method for mathematical programs with equilibrium constraints. *Mathematical programming*, 85(1):107–134, 1999.

[FL08] Liming Feng and Vadim Linetsky. Pricing options in jump-diffusion models: an extrapolation approach. *Operations Research*, 56(2):304–325, 2008.

[FLMN10] Liming Feng, Vadim Linetsky, José Luis Morales, and Jorge Nocedal. On the solution of complementarity problems arising in American options pricing. *Optimization Methods and Software*, Forthcoming, 2010.

[FLP98] Masao Fukushima, Zhi-Quan Luo, and Jong-Shi Pang. A globally convergent sequential quadratic programming algorithm for mathematical programs with linear complementarity constraints. *Computational Optimization and Applications*, 10(1):5–34, 1998.

[FLRS06] Roger Fletcher, Sven Leyffer, Danny Ralph, and Stefan Scholtes. Local convergence of sqp methods for mathematical programs with equilibrium constraints. *SIAM Journal on Optimization*, 17(1):259–286, 2006.

[FP03a] Francisco Facchinei and Jong-Shi Pang. *Finite-dimensional variational inequalities and complementarity problems*, volume 2. Springer, 2003.

[FP03b] Francisco Facchinei and Jong-Shi Pang. *Finite-dimensional variational inequalities and complementarity problems*, volume 1. Springer, 2003.

[Gib91] Rajna Gibson. *Option valuation: Analyzing and pricing standardized option contracts*. McGraw-Hill New York, 1991.

[GVL12] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[Hes93] Steven L Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies*, 6(2):327–343, 1993.

[HIK02] Michael Hintermüller, Kazufumi Ito, and Karl Kunisch. The primal-dual active set strategy as a semismooth newton method. *SIAM Journal on Optimization*, 13(3):865–888, 2002.

[HP98] Jacqueline Huang and Jong-Shi Pang. Option pricing and linear complementarity. *Journal of Computational Finance*, 2(3):31–60, 1998.

[HP00] Jacqueline Huang and Jong-Shi Pang. A mathematical programming with equilibrium constraints approach to the implied volatility surface of American options. *Journal of Computational Finance*, 4(1):21–56, 2000.

[Hul09] John Hull. *Options Futures and Other Derivatives.* Prentice Hall, Upper Saddle River, New Jersey, seventh edition, 2009.

[HW87] John Hull and Alan White. The pricing of options on assets with stochastic volatilities. *The journal of finance*, 42(2):281–300, 1987.

[IT04] Samuli Ikonen and Jari Toivanen. Operator splitting methods for american option pricing. *Applied mathematics letters*, 17(7):809–814, 2004.

[JLL90] Patrick Jaillet, Damien Lamberton, and Bernard Lapeyre. Variational inequalities and the pricing of american options. *Acta Applicandae Mathematica*, 21(3):263–289, 1990.

[JS87] Herb Johnson and David Shanno. Option pricing when the variance is changing. *Journal of Financial and Quantitative Analysis*, 22(2):143–151, 1987.

[Kou02] Steven G Kou. A jump-diffusion model for option pricing. *Management Science*, 48(8):1086–1101, 2002.

[LLCN06] Sven Leyffer, Gabriel López-Calva, and Jorge Nocedal. Interior methods for mathematical programs with complementarity constraints. *SIAM Journal on Optimization*, 17(1):52–77, 2006.

[LO97a] Ronald Lagnado and Stanley Osher. Reconciling differences. *Risk-London-Risk magazine limited-*, 10:79–83, 1997.

[LO97b] Ronald Lagnado and Stanley Osher. A technique for calibrating derivative security pricing models: numerical solution of an inverse problem. *Journal of computational finance*, 1(1):13–25, 1997.

[LPR96] Zhi-Quan Luo, Jong-Shi Pang, and Daniel Ralph. *Mathematical programs with equilibrium constraints.* Cambridge University Press, 1996.

[LR76] Henry A Latane and Richard J Rendleman. Standard deviations of stock price ratios implied in option prices. *The Journal of Finance*, 31(2):369–381, 1976.

[Mer73] Robert C Merton. Theory of rational option pricing. *Bell Journal of Economics and Management Science*, 4(1):141–183, 1973.

[Mer76] Robert C Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3(1-2):125–144, 1976.

[MNS08] José Luis Morales, Jorge Nocedal, and Mikhail Smelyanskiy. An algorithm for the fast solution of symmetric linear complementarity problems. *Numerische Mathematik*, 111(2):251–266, 2008.

[MT97] William R Melick and Charles P Thomas. Recovering an asset's implied pdf from option prices: an application to crude oil during the gulf crisis. *Journal of Financial and Quantitative Analysis*, 32(1), 1997.

[Mur88] Katta G Murty. *Linear complementarity, linear and nonlinear programming.* Heldermann Berlin, 1988.

[PC85] Jong-Shi Pang and Ramaswamy Chandrasekaran. Linear complementarity problems solvable by a polynomially bounded pivoting algorithm. In *Mathematical Programming Essays in Honor of George B. Dantzig Part II*, pages 13–27. Springer, 1985.

[PHR91] Jong-Shi Pang, Shih-Ping Han, and Narayan Rangaraj. Minimization of locally lipschitzian functions. *SIAM Journal on Optimization*, 1(1):57–82, 1991.

[RFNP11] Daniel P Robinson, Liming Feng, Jorge M Nocedal, and Jong-Shi Pang. Subspace accelerated matrix splitting algorithms for bound-constrained quadratic programming and linear complementarity problems, 2011.

[Rub94] Mark Rubinstein. Implied binomial trees. *Journal of Finance*, 49(3):771–818, 1994.

[Shr04] Steven E Shreve. *Stochastic calculus for finance II: Continuous-time models*, volume 11. Springer, 2004.

[SS00] Holger Scheel and Stefan Scholtes. Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity. *Mathematics of Operations Research*, 25(1):1–22, 2000.

[Tan04] Peter Tankov. *Financial modelling with jump processes.* CRC Press, 2004.

[WHD95] Paul Wilmott, Sam Howison, and Jeff Dewynne. *The Mathematics of Financial Derivatives.* Cambridge University Press, Cambridge, UK, 1995.

[ZFV98] Robert Zvan, Peter A Forsyth, and Kenneth R Vetzal. Penalty methods for american options with stochastic volatility. *Journal of Computational and Applied Mathematics*, 91(2):199–218, 1998.