JYVÄSKYLÄ STUDIES IN COMPUTING
180

Santtu Salmi

Numerical Methods for Pricing Options under Jump-Diffusion Processes

JYVÄSKYLÄN YLIOPISTO

JYVÄSKYLÄ STUDIES IN COMPUTING 180

Santtu Salmi

Numerical Methods for Pricing Options under Jump-Diffusion Processes

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi yliopiston Agora-rakennuksen auditoriossa 2
joulukuun 14. päivänä 2013 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in building Agora, Auditorium 2, on December 14, 2013 at 12 o'clock noon.

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2013

Numerical Methods for Pricing Options under Jump-Diffusion Processes

JYVÄSKYLÄ STUDIES IN COMPUTING 180

Santtu Salmi

Numerical Methods for Pricing Options under Jump-Diffusion Processes

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2013

Editors
Timo Hämäläinen
Department of Department of Mathematical Information Technology, University of Jyväskylä
Pekka Olsbo, Sini Tuikka
Publishing Unit, University Library of Jyväskylä

Cover picture by Nelli Salmi

URN:ISBN:978-951-39-5514-4
ISBN 978-951-39-5514-4 (PDF)

ISBN 978-951-39-5513-7 (nid.)
ISSN 1456-5390

Jyväskylä University Printing House, Jyväskylä 2013

ABSTRACT

Salmi, Santtu
Numerical Methods for Pricing Options under Jump-Diffusion Processes
Jyväskylä: University of Jyväskylä, 2013, 50 p.(+included articles)
(Jyväskylä Studies in Computing
ISSN 1456-5390; 180)
ISBN 978-951-39-5513-7 (nid.)
ISBN 978-951-39-5514-4 (PDF)
Finnish summary
Diss.

This dissertation deals with the numerical solution of partial integro-differential equations (PIDEs) derived for the pricing of options under jump-diffusion processes. The addition of jumps into the option pricing model arguably leads to a more realistic account of real-world market behavior. However, due to the non-locality of these jump terms, the discretized systems become dense. Special methods and algorithms are necessary to solve these systems efficiently. Iterative methods and special time discretizations are employed to circumvent the inversion of a full matrix. Full matrix-vector multiplications are still necessary, which can be computed efficiently by employing Fourier transforms or recursion formulas.

Keywords: option pricing, jump-diffusion, numerical methods, PIDE

Author

Santtu Salmi
Department of Mathematical Information Technology
University of Jyväskylä, Finland

Supervisors

Dr. Jari Toivanen
Department of Mathematical Information Technology
University of Jyväskylä, Finland

Institute for Computational and Mathematical Engineering
Stanford University, USA

Prof. Raino A. E. Mäkinen
Department of Mathematical Information Technology
University of Jyväskylä, Finland

Reviewers

Prof. Abdul Q. M. Khaliq
Department of Mathematical Sciences
Middle Tennessee State University, USA

Prof. Hongtao Yang
Department of Mathematical Sciences
University of Nevada, USA

Opponent

Prof. Cornelis W. Oosterlee
Delft Institute of Applied Mathematics
Delft University of Technology, The Netherlands

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to Jari Toivanen. I feel privileged to have had such a talented supervisor, who not only possesses expert knowledge in his research areas, but who also has sacrificed innumerable hours of his time to patiently answer my questions and provide spontaneous guidance. Thank you Jari, I cannot imagine anyone who could have done a better job.

I wish to thank Lina von Sydow for her hard work on articles [PIV] and [PV] of this dissertation, and also for welcoming me to visit the beautiful city and university of Uppsala. I am grateful to Raino Mäkinen for providing invaluable feedback on the dissertation, and for always being available to resolve any practical issues.

I would like to express my gratitude to the Väisälä Foundation of the Finnish Academy of Science and Letters for providing me with two sizable scholarships that have, in large part, enabled me to carry out my doctoral studies. I am also grateful to Aino Sallinen, the former rector of University of Jyväskylä for providing me with yet another critically important scholarship. Furthermore, I would like to thank the COMAS doctoral program and the former vice-rector Matti Manninen for the granted funding to travel to Stanford University and various conferences.

For carefully reviewing the dissertation manuscript I wish to thank Abdul Q. M. Khaliq and Hongtao Yang. Moreover, I am thankful for the insightful comments and remarks made by the anonymous referees involved in reviewing the published papers of this dissertation. I am grateful to Cornelis W. Oosterlee for agreeing to be my opponent, and for inviting me to the very enjoyable workshop in Leiden. Finally, I wish to thank all my friends, family and colleagues for discussions, encouragement, and inspiration.

Jyväskylä, November 2013
Santtu Salmi

LIST OF FIGURES

LIST OF TABLES

CONTENTS

LIST OF INCLUDED ARTICLES

PI S. Salmi and J. Toivanen. An iterative method for pricing American options under jump-diffusion models. *Applied Numerical Mathematics, Volume 61, Issue 7, 2011, pp. 821-831*, 2011.

PII S. Salmi and J. Toivanen. Comparison and survey of finite difference methods for pricing American options under finite activity jump-diffusion models. *International Journal of Computer Mathematics, Volume 89, Issue 9, pp. 1112-1134*, 2012.

PIII S. Salmi and J. Toivanen. Robust and efficient IMEX schemes for option pricing under jump-diffusion models. *Submitted for publication*, 2013.

PIV S. Salmi, J. Toivanen and L. von Sydow. Iterative methods for pricing American options under the Bates model. *Proceedings of 2013 International Conference on Computational Science, Procedia Computer Science Series, Volume 18 pp. 1136-1144.*, 2013.

PV S. Salmi, J. Toivanen and L. von Sydow. An IMEX-scheme for pricing options under stochastic volatility models with jumps. *Submitted for publication*, 2013.

1 INTRODUCTION

The first recorded account of an investment mimicking an option contract was mentioned in the book *Politics* [8] by Aristotle in the fourth century B.C. A man named Thales of Miletus observed the weather patterns and predicted a huge olive harvest. Thales understood that due to the rising demand in the use of olive presses (olives could not be stored at that time), he could make a big profit if he owned the olive presses in the area. However, since he did not have the money to buy the olive presses, instead, he had a brilliant idea. Thales invested a small amount of money to secure the use of the olive presses in the region during the harvest. As he predicted, harvest was plentiful and he made a big fortune by selling the rights to use the olive presses.

This kind of a financial contract is understood today as an option, where the owner of the contract has the option, but not the obligation, to exercise the option. Thales had the option of using the olive presses himself, exercising the option, or selling the option contract for profit. It was possible, that due to an unforeseeable disaster the olive harvest would have been destroyed. Therefore, Thales ran the risk of losing his initial small investment. Alternatively, the harvest could have been merely average, in which case Thales would have probably just covered his expenses and made no profit. This was an example of an option holder speculating on the price of the underlying asset, in this case, the right to use an olive press. The option contract seller or *writer*, here the olive press owners, had the guaranteed fixed profit of Thales' initial investment even in the case of a disaster, but the writers did not benefit from the higher olive press rental prices generated by increasing demand. The option writers were hoping the underlying asset price to remain at, or go below, the price that Thales paid them. This amount is called the strike price of the option.

In contrast to speculation, another, arguably more important purpose for option contracts is to hedge against risk. Let us consider a typical example of a commercial farmer. The market price of wheat, rice and other crops fluctuate according to supply and demand. The decision to plant a specific type of crop needs to be made a season in advance, and after planting the farmer is committed to this choice. Assume that the farmer decides that planting wheat is a good

idea. If the price of wheat goes up during this time, he makes an unexpected profit. If the price goes down, however, he could be ruined. To protect himself from this risk, the farmer can enter into a financial contract, where he promises to deliver a certain amount of wheat with a fixed price at a certain date in the future. This is known as a futures contract. He is no longer exposed to the price fluctuations in the markets. However, he also gives up the chance to make extra profits if the price of wheat is higher at harvest, as he is *obliged* to sell under the conditions written in the futures contract. To retain the chance to make extra profits at harvest, the farmer can instead enter into a financial contract where he has the *option* to *sell* under the specified conditions, but not the obligation. In financial terminology this is called a *put option*. Similarly, the option to *buy* the underlying asset under the specified conditions is called a *call option*. Put options are often seen as a form of insurance. However, while you cannot insure someone else's house, you can buy an option on an asset you do not own. Also, unlike regular insurance there is no limit on how many options you can own. This enables put options to be used also in a speculative manner.

While a futures contract is binding on both parties, an option contract is binding only on the writer. The writer of the option is obliged to buy or sell the underlying asset under the specified conditions in case the option holder wants to exercise the option. An option contract is therefore an asymmetric agreement between the parties, and it clearly always has a positive value for the holder. The option holder has to pay the writer for agreeing to the obligation imposed by the option contract. In other words, the holder pays the writer for carrying the risks associated with the fluctuations of the underlying asset. This leads to the fundamental question: how much should the option holder pay the writer?

In 1973, for the first time this question was answered in a rigorous manner in the pioneering papers by Black and Scholes [10], and Merton [52]. The Black-Scholes model was certainly not the first attempt to give a fair price to an option, but it was the first time a completely tractable solution was given. The model consists of a random process, namely geometric Brownian motion, and a deterministic drift. As a side note, already as early as the year 1900 random walk, a discrete version of Brownian motion, was proposed to model asset price movement by Bachelier in his thesis [6]. Only two parameters are needed as additional inputs in the Black-Scholes model, which are the risk-free interest rate, a quantity that can be approximated by observing very low risk assets such as government bonds, and the market volatility, also a quantity that can be estimated from financial time series.

Black, Scholes, and Merton constructed an investment portfolio that contains both a written option contract (also known as a short position on an option) and a dynamically adjusted amount of the underlying asset (e.g. stock). If the amount of the underlying asset is continuously adjusted in the portfolio in just the right way, a technique called delta-hedging, the value of the portfolio remains unchanged under any continuous changes in the underlying asset. By applying Itô's lemma to the portfolio, a kind of a chain-rule in stochastic calculus, the randomness in the portfolio vanishes. Since the portfolio is now riskless,

it must have a rate of return equal to any other riskless instrument. This is the so-called no-arbitrage argument. This line of reasoning results in the celebrated Black-Scholes partial differential equation (PDE), which satisfies the fair price of a European-style option contract. This PDE admits an analytic solution, the so-called Black-Scholes *formula*.

The Black-Scholes PDE undeniably revolutionized the field of finance. Before, option prices were settled individually between two parties, and the price was based merely on intuition and negotiation. After, a fair price of an option could be obtained via the Black-Scholes formula. In 1973 option trading boomed when the Chicago Board Options Exchange started trading options in an open exchange. In recent years, however, the limitations of the Black-Scholes model have come under more attention.

In this dissertation we price options under models that are more general, and presumably more realistic, than the Black-Scholes model. In particular, jump-diffusion models are our central focus. The analytical solution to the classical Black-Scholes model is known, but this is usually not the case for more general models. Hence, numerical solution is necessary. Numerical solution methods for option pricing can be roughly divided into three categories: numerical solution of PDEs, Monte Carlo simulation, and numerical integration techniques. Here we adopt the numerical solution of PDEs approach. Each numerical solution method has their particular strengths and weaknesses. For example, PDE methods are known to be strong for low dimensional problems, especially for American-style option contracts. Numerical integration techniques are extremely powerful for European-style option contracts, and naturally also for contracts that can be reduced to the European option pricing problem. Monte Carlo simulation is preferable for high dimensional problems (e.g. basket options), and also for some exotic contracts where an explicit PDE might be difficult to derive.

2 FINANCIAL OPTION CONTRACTS

In the following we discuss different styles of option contracts. The most commonly traded options include European and American options[1], which are collectively known as vanilla options. Conversely, options with nonstandard features are generally known as exotic options, although there is no clear distinctive definition between the two categories. Typically exotic options are so-called over-the-counter (OTC) options, that is, they are traded and negotiated between two private parties.

The defining characteristics of an option contract are: the governing underlying asset(s), strike price, expiry date, payoff function, and the option contract style. The option may be exercised at specified dates, depending on the style of the option contract, at or before the expiry date. In case of a call option, the option holder has the right to *buy* the underlying asset with the strike price, and similarly the holder of a put option has the option to *sell* the underlying asset with the strike price. Additionally, option contracts include settlement terms, which specify for example, whether the writer must deliver the actual asset on exercise, or may simply tender the equivalent cash amount (cash settlement). For instance, the standard practice for options depending on a stock is to deliver the equivalent cash amount on exercise instead of the actual underlying stock. In this case, exercising the option yields a payoff, which for vanilla options is the positive difference between the strike price and the value of the underlying asset.

2.1 European Options

European style option contracts are usually index and commodity options. A holder of a European-style option has the right to buy or sell the underlying asset with the strike price *at* the expiry date. In other words, the right to exercise a European option is only at the expiry date, at a single predefined point in time.

[1] European and American options are traded everywhere in the world. The terminology originated simply upon which continent these styles of options where first traded in.

In practice, a European option is automatically exercised if it has any value at expiry. Let us denote the value of the underlying asset at time $t \in [0, T]$ by S_t, where T is the expiry time. Now, the payoff function $g(S_t)$ of a European style *call* option is given by

$$g(S_T) = \max\{S_T - K, 0\}, \tag{1}$$

where K is the strike price of the option. Clearly the option has positive value only if the value of the underlying asset S_t at time $t = T$ is greater than the strike price K. Otherwise, the holder of the option is better off buying the underlying asset from the market, and in this case the option contract was a worthless investment. Similarly, the payoff function $g(S_t)$ of a European style *put* option is given by

$$g(S_T) = \max\{K - S_T, 0\}. \tag{2}$$

The payoff functions for $K = 100$ are plotted in Figure 1; also, option prices given by the Black-Scholes formula, under parameters given in Table 1, are included to illustrate the evolution of the option price (backwards) in time. Notice, however, that the price evolution is different for models other than the Black-Scholes model. These models will be discussed further in the next chapter.

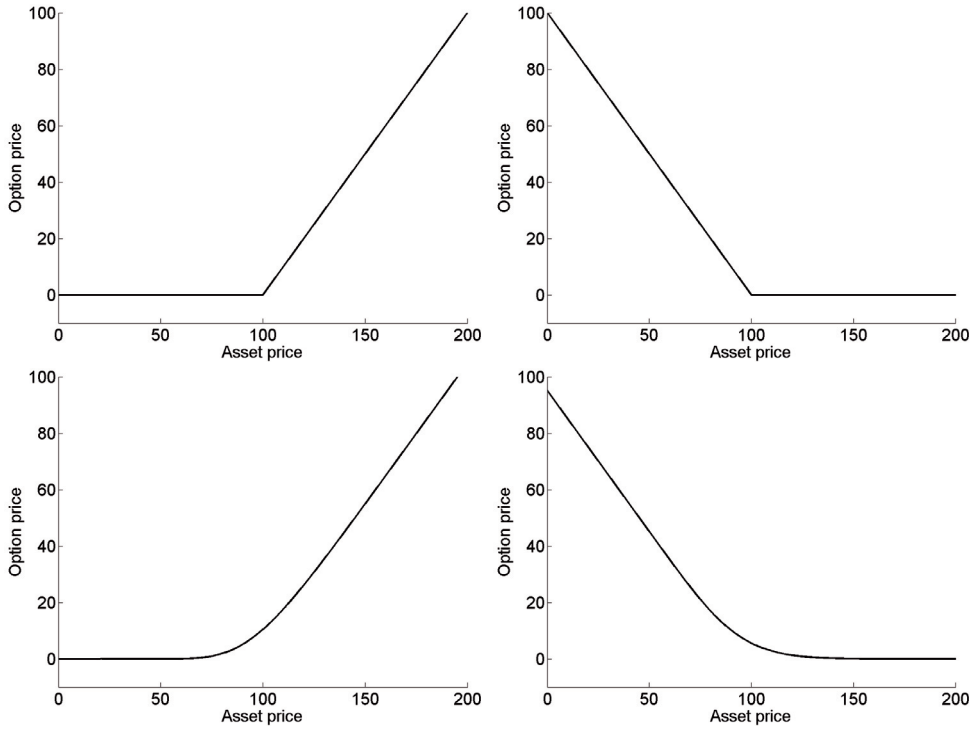

FIGURE 1 Payoff functions for European options at expiry: call option (Top Left), put option (Top Right), and option prices given by the Black-Scholes formula at one year before expiry: call option (Bottom Left), put option (Bottom Right).

TABLE 1 Black-Scholes parameters

Strike price K	100
Expiry time T	1
Risk-free interest rate r	0.05
Volatility σ	0.2

2.2 American Options

Contrary to European options, American options can be exercised at any point in time before expiry. Options that have this type of feature are known as early-exercise options. However, American options are unique in that they can be exercised at any point in time, while other early-exercise options typically allow early exercise at some predefined times before expiry. Due to this early exercise possibility, the value of an American option is always greater than the value of a corresponding European option.

The payoff function $g(S_t)$ of an American *call* option at time $t \in [0, T]$ is given by

$$g(S_t) = \max\{S_t - K, 0\}. \tag{3}$$

Since exercise is possible at any time $t \in [0, T]$, the natural question is: when should one exercise the option? Exercising if $S_t \leq K$ is certainly not sensible since the payoff is zero. If $S_t > K$ the answer is not so clear anymore. The payoff is positive, but by exercising the option, the holder also forgoes the possibility of future profits. This is known as the time value of the option. The more time there is left before expiry, the higher the probability of large changes in the underlying. However, by exercising the option, the holder also eliminates the risk of losing profits through the depreciation of the underlying asset. In fact, the early exercise of an American option is an optimal decision problem, and the region in the underlying asset space where exercise is optimal is called the early exercise region. This region changes in time, and it is characterized by the early exercise boundary, which for a simple model is a point in the asset price space moving with time. American options are typically stock, equity, and commodity options, which are often the most traded options in an exchange.

2.3 Other styles of Options

Here we present a brief overview of some well-known non-vanilla options. A Bermudan option, for example, is a simple variant of vanilla options. A Bermudan option has the same payoff function as European and American options, as given by (3) for a call option. However, Bermudan options can be exercised only

at a predefined discrete set of times. Therefore, Bermudan options are an intermediate between European and American options. Indeed, the name Bermudan option is a pun, derived from the fact that Bermuda is geographically located between Europe and America. From a mathematical perspective, pricing a Bermudan option can be reduced to pricing several European options (one for each exercise time of the Bermudan option). This is a significant advantage, since European options can often be priced with analytic or semi-analytic pricing methods, whereas American options typically require numerical solution.

Options with unconventional payoff functions are known as exotic options. A simple exotic option is a basket option, where the payoff value depends on the weighted average of several underlying assets. Other exotic options include, but are not limited to: lookback options, Asian options, binary options, and barrier options. A lookback option is a path dependent option where the option holder has the right to buy (sell) the underlying asset at its lowest (highest) price over the lifetime of the option. Similarly, the value of an Asian option is determined by the *average* value of the underlying asset over the lifetime of the option. A binary option, as its name suggests, simply pays a fixed amount or nothing at all, depending on the value of the underlying asset at expiry. Barrier options can be divided into two categories: the option springs into existence after the underlying asset breaches a barrier, or conversely, the option is extinguished if the barrier is breached. The first type of barrier options are known as *up and in*, *knock-in*, or *down and in* options. The second type of barrier options are known as *up and out*, *knock-out*, or *down and out* options. There are also so-called double barrier options that have two distinct barriers. If the underlying asset crosses either one of these barriers, the option either springs into existence (knock-in) or is extinguished (knock-out). Clearly, barrier options are always cheaper than a similar option without a barrier, since the existence of the option is conditional.

In addition to the exotic options presented above, there is a wide variety of other exotic options. However, in many cases the price of an exotic option can be derived from the price of a vanilla option. Thus, the pricing of European and American options is of fundamental importance. In the remainder of this dissertation we will focus on the pricing of European and American options.

3 FINANCIAL MARKET MODELS

Even at the time of the introduction of the Black-Scholes model, the constant market volatility assumption and continuity of sample paths were known to be simplifications. Fitting empirically observed option prices into the Black-Scholes model implies a volatility distribution pattern so commonly observed, that it has been given its own name, the volatility smile. The implied volatility is typically lower near the strike price, and higher far from the strike price of the option. Thus, plotting the implied volatility against the strike price yields a smile like curve. In the following, implied volatility is always assumed to be given by the Black-Scholes model. It is known that the implied volatility becomes more pronounced for options with short maturities, or in other words, for options that are about to expire. The volatility smile can be seen as empirical evidence suggesting that the Black-Scholes model is misspecified. Indeed, the use of the Black-Scholes formula by financial practitioners was described in [60] as using "the wrong number in the wrong formula to get the right price". Under the Black-Scholes model options can be perfectly hedged by a replicating strategy, which holds only the underlying asset and a risk-free bond. Thus, options in this (complete market) setting are, perplexingly, redundant assets.

Other, more subtle concerns have also risen. For example, stock returns are known to exhibit heavier tails than the log-normal distribution implied by the Black-Scholes model. Also, the return distribution is typically skewed, where extreme negative movements are more likely than extreme positive movements. The usual modification to explain heavy tails is the introduction of jumps into the model. Merton proposed log-normally distributed jumps in [53] already in 1976. Another well-known jump-diffusion model is the Kou model, where jumps are drawn from a log-double-exponential distribution [46]. This offers a higher probability for extreme events and allows asymmetry in the jump distribution. The possibility of jumps is particularly important for options with short maturities, since the probability of a purely Brownian motion type process experiencing a large sudden movement is almost zero. However, large sudden changes are occasionally recorded, and the pronounced implied volatility smile near expiry suggests that markets expect these kinds of movements.

The aforementioned limitations of the Black-Scholes model are not only a modeling issue. Blind trust in the Black-Scholes model can lead to the underestimation of risk. If real-world assets are assumed to follow a geometric Brownian motion, then banks can write an option with the Black-Scholes price and hedge their position perfectly using the delta-hedging technique. They can charge a small fee on top of the Black-Scholes price to make a small profit with no risk[1]. Now assume, instead, that every 10 years there is a market crash, where asset prices make a big sudden jump downward. In this scenario the bank makes small steady profits for 10 years, but on the 10th year these profits are wiped out. The moral hazard is apparent if bankers receive their bonuses during the good years, but ultimately are not themselves financially exposed to the market crash. It is not the case that financial practitioners do not know about the limitations of the Black-Scholes model, rather, it might be in their own interest to remain optimistic.

In the following we describe and formulate some well-known market models. The usual modifications to the Black-Scholes model include stochastic volatility and jumps.

3.1 Black-Scholes Model

In the Black-Scholes model there are two assets. One is a riskless asset (risk-free bond) with price β_t, described by the ordinary differential equation

$$d\beta_t = r\beta_t dt, \tag{4}$$

where r is the instantaneous (continuously compounded) interest rate for lending or borrowing. Setting $\beta_0 = 1$ gives $\beta_t = e^{rt}$ for $t \geq 0$. The dynamics of the second asset (the risky asset) are given by the following stochastic differential equation (SDE)

$$dS_t = \mu S_t dt + \sigma S_t dW_t, \tag{5}$$

where S_t is the value of the underlying asset at time t, and μ and $\sigma \geq 0$ are constants that determine the magnitude of drift and volatility of the process, respectively. Above, W_t is the Wiener process, also known as the standard Brownian motion. A process that satisfies the SDE (5) is said to follow a *geometric* Brownian motion. The SDE (5) is a shorthand notation of the following integral equation

$$S_t = S_0 + \mu \int_0^t S_u du + \sigma \int_0^t S_u dW_u. \tag{6}$$

The integral on the left is an ordinary Lebesgue integral, while the integral on the right is a stochastic integral. This can be identified from the integrator, which for a stochastic integral is a stochastic process, in this case the Wiener process W_u. The SDE (5) is also often written in the equivalent form

$$\frac{dS_t}{S_t} = \mu dt + \sigma dW_t, \tag{7}$$

[1] To be precise, here it is also necessary to assume that there is no hedging risk.

where dS_t/S_t is the infinitesimal return (on investment).

There are multiple ways to derive the Black-Scholes PDE. Here we follow the intuitive presentation given by Hull in [35], which is based on the original paper by Black and Scholes [10]. The following assumptions are made under the Black-Scholes model:

1. The underlying asset S_t satisfies the SDE (5), with constant μ and σ.
2. The short selling of assets is permitted.
3. There are no transactions costs or taxes, and assets are perfectly divisible.
4. The underlying asset pays no dividend.
5. There are no riskless arbitrage opportunities.
6. Trading is continuous.
7. The risk-free rate of interest r is constant.

Some of these assumptions can be relaxed. For example, the risk-free interest rate r or volatility σ can be assumed to be non-constant deterministic quantities (e.g. the so-called local volatility model). Alternatively, r and σ can be stochastic processes themselves, which leads to stochastic interest rate or stochastic volatility models. Also, dividend payment is often included by approximating it with an additional drift term. However, in the following derivation we assume that the assumptions given above hold.

Assume $v(S_t, t)$ is the *price* of an option contract written on the underlying asset S_t. The payoff of the option at expiry $v(S_T, T)$ is known. For example, the payoff function at expiry for a European call option was given in (1). To find the value of $v := v(S_t, t)$ at earlier times t, we need to know how v evolves as a function of S_t and t. For this purpose, we need a fundamental result in stochastic calculus known as the Itô's lemma [41].

Intuitively Itô's lemma can be thought of as the stochastic equivalent of the chain-rule. In its simplest form Itô's lemma states that for a twice differentiable scalar function $f(x, t)$ of two real variables x and t the following equation holds[2]

$$df(X_t, t) = \left(\frac{\partial f}{\partial x}\mu + \frac{\partial f}{\partial t} + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}\sigma^2\right)dt + \sigma\frac{\partial f}{\partial x}dW_t, \tag{8}$$

where X_t is a so-called Itô drift-diffusion process given by

$$dX_t = \mu dt + \sigma dW_t. \tag{9}$$

Now, recall that v is a function of S_t, which follows a *geometric* Brownian motion. Thus, by applying Itô's lemma on v we obtain the following equation

$$dv(S_t, t) = \left(\frac{\partial v}{\partial S_t}\mu S_t + \frac{\partial v}{\partial t} + \frac{1}{2}\frac{\partial^2 v}{\partial S_t^2}\sigma^2 S_t^2\right)dt + \sigma S_t\frac{\partial v}{\partial S_t}dW_t. \tag{10}$$

[2] For an informal derivation of Itô's lemma see [35], where the proof is sketched using Taylor polynomials. A formal proof can be found in [56], for example. Here it is sufficient to take a leap of faith and take (8) for granted.

Let us write the discrete versions of SDEs (5) and (10) given by

$$\Delta S_t = \mu S_t \Delta t + \sigma S_t \Delta W_t, \tag{11}$$

and

$$\Delta v = \left(\frac{\partial v}{\partial S_t} \mu S_t + \frac{\partial v}{\partial t} + \frac{1}{2} \frac{\partial^2 v}{\partial S_t^2} \sigma^2 S_t^2 \right) \Delta t + \sigma S_t \frac{\partial v}{\partial S_t} \Delta W_t, \tag{12}$$

where Δv, ΔS_t, and ΔW_t are the changes in v, S_t, and W_t in a small time interval Δt. Now, let us construct the so-called delta-hedge portfolio, which consists of a written option $-v$ and $\frac{\partial v}{\partial S_t}$ amount of the underlying asset S_t at time t. This portfolio is delta-neutral at time t, that is, it is hedged against any continuous changes in the underlying asset at time instant t. Let us denote the value of the portfolio by Π, which by definition is

$$\Pi = -v + \frac{\partial v}{\partial S_t} S_t. \tag{13}$$

The change $\Delta\Pi$ in the value of the portfolio in the time interval Δt is given by

$$\Delta\Pi = -\Delta v + \frac{\partial v}{\partial S_t} \Delta S_t. \tag{14}$$

Substituting (11) and (12) into the equation (14) above yields

$$\Delta\Pi = \left(-\frac{\partial v}{\partial t} - \frac{1}{2} \frac{\partial^2 v}{\partial S_t^2} \sigma^2 S_t^2 \right) \Delta t. \tag{15}$$

Most important, notice how the terms with μ and W_t have vanished. This is the key breakthrough of Black and Scholes in [10] and Merton in [52]. The portfolio no longer has any source of uncertainty, hence, it is now riskless. We have constructed a deterministic hedging strategy for the written option. Previously it was assumed that there are no riskless arbitrage possibilities. This implies that the portfolio must have a rate of return equal to any other riskless instrument, such as the risk-free bond in (4). Therefore, the following equation holds

$$\Delta\Pi = r\Pi\Delta t, \tag{16}$$

where r is the risk-free interest rate. Now, by substituting (13) and (15) into (16) we have

$$\left(\frac{\partial v}{\partial t} + \frac{1}{2} \frac{\partial^2 v}{\partial S_t^2} \sigma^2 S_t^2 \right) \Delta t = r \left(v - \frac{\partial v}{\partial S_t} \right) \Delta t. \tag{17}$$

By rearranging the terms and dividing by Δt we obtain

$$\frac{\partial v}{\partial t} + \frac{1}{2} \sigma^2 S_t^2 \frac{\partial^2 v}{\partial S_t^2} + r S_t \frac{\partial v}{\partial S_t} - r v = 0, \tag{18}$$

which is the Black-Scholes PDE[3]. This PDE governs the evolution of the price $v(S_t, t)$ of the option contract backwards in time, where the final (initial) price is known at time $t = T$ to be equal to the payoff function of the option.

[3] Notice, that the discretization approach used here implicitly assumes that the portfolio Π is self-financing. The derivation of the Black-Scholes PDE in a continuous setting requires this additional assumption.

3.2 Stochastic Volatility Models

The constant volatility assumption of the Black-Scholes model is arguably one of its most prominent shortcomings. In real world markets volatility tends to cluster. In other words, large movements tend to be followed by large movements and vice versa. Also, heavy tails of the return distribution can partly be accounted for by relaxing the constant volatility assumption, particularly for larger time intervals. Here we consider models where the volatility is presumed to be a stochastic process. Therefore, we have a stochastic process to model the risky asset, and another stochastic process to model the volatility of the former process. The so-called local volatility model is a special case, which we will not entertain further here, where the volatility is a deterministic time dependent process. Markets under local volatility are complete, whereas stochastic volatility leads to an incomplete market setting. Put differently, the hedging of options under stochastic volatility is no longer a triviality, on the contrary, it becomes an inverse problem.

The underlying asset S_t under stochastic volatility models follows the SDE

$$dS_t = \mu S_t dt + \sigma_t S_t dW_t, \tag{19}$$

where σ_t is the volatility process. Various models have been proposed for the volatility process; see [7, 30, 36, 62, 64] for example. Generally, the volatility process is written in the asset price model as $\sigma_t = f(V_t)$, for some positive function f. Here, it is sufficient to write $\sigma_t := \sqrt{V_t}$, where V_t is the variance of σ_t. The SDE now reads

$$\frac{dS_t}{S_t} = \mu dt + \sqrt{V_t} dW_t. \tag{20}$$

The (square-root) volatility process V_t is typically taken to be an Itô drift-diffusion process, that is, it satisfies an SDE driven by a Brownian motion with a drift. This Brownian motion is distinct from the Brownian motion driving the asset price process, however, a correlation between the two Brownian motions is often included.

An important feature that many volatility models share is *mean reversion*. For instance, the Ornstein-Uhlenbeck process [69] given by

$$dV_t = a(b - V_t)dt + \psi dW_t \tag{21}$$

is mean reverting with long term mean level b, rate of mean reversion a and volatility of volatility ψ. Another popular mean reverting process is the Cox-Ingersoll-Ross (CIR) process [15], which reads

$$dV_t = a(b - V_t)dt + \psi\sqrt{V_t} dW_t. \tag{22}$$

The mean reversion eventually pulls the process back towards the long term mean level (imagine a rubber band attached to the mean level). The volatility model proposed by Hull and White [36] in 1987 follows a geometric Brownian

motion, which is *not* mean reverting[4]. However, observations suggest that real world market volatility is mean reverting [26].

The SDEs used to model volatility are often the same models that are used for interest rate modeling, where mean reversion is also an important property [15, 37, 70]. For example, in the Heston stochastic volatility model [30] the volatility is modeled by the CIR process given in equation (22). The Heston stochastic volatility model reads

$$\begin{aligned} \frac{dS_t}{S_t} &= \mu dt + \sqrt{V_t} dW_t, \\ dV_t &= a(b - V_t)dt + \psi\sqrt{V_t} dZ_t, \end{aligned} \tag{23}$$

where W_t and Z_t are correlated Wiener processes. The correlation can be written as

$$Z_t = \rho W_t + \sqrt{1 - \rho^2}\hat{Z}_t, \tag{24}$$

where $\hat{Z}_t$ is a Wiener process independent of W_t, and $\rho \in [-1, 1]$ is the instantaneous correlation coefficient. If the parameters satisfy the so-called Feller condition

$$2ab \geq \psi^2, \tag{25}$$

then the volatility (CIR) process V_t is strictly positive. The Heston model is a popular stochastic volatility model most likely because it admits a closed-form solution for European-style options. The solution is given by a complex integral, which needs to be evaluated numerically. The pricing of options under stochastic volatility is somewhat more involved, since the variable V_t leads to an extra degree of freedom.

The PDE that determines the evolution of the price $v := v(S_t, V_t, t)$ of an option under the Heston model is given by

$$\begin{aligned} &\frac{\partial v}{\partial t} + \frac{1}{2}V_t S_t^2 \frac{\partial^2 v}{\partial S_t^2} + \rho\psi V_t S_t \frac{\partial^2 v}{\partial S_t V_t} + \frac{1}{2}\psi^2 V_t \frac{\partial^2 v}{\partial V_t^2} + \\ &\quad rS_t \frac{\partial v}{\partial S_t} + (a(b - V_t) - \beta(S_t, V_t, t))\frac{\partial v}{\partial V_t} - rv = 0, \end{aligned} \tag{26}$$

where β is the price of volatility risk. The reason for this term is that in reality most investors are found to be risk averse in experimental settings [32]. Moreover, Lamoureux and Lastrapes find evidence from observed option prices that the efficient-market hypothesis and investor risk-neutrality cannot hold simultaneously [49]. Often β is assumed zero, however. Then the price is said to be given under the risk-neutral measure, i.e., under the assumption that investors are risk-neutral. In the following formulations we will assume $\beta = 0$. In models with jumps, in particular, the price of volatility risk term β is often found to be unnecessary [4, 57].

[4] There is a risk of confusion here, since Hull and White are associated with a stochastic volatility model and an interest rate model. The interest rate model *is* mean reverting. To add to the confusion, the Vasicek and Hull-White interest rate models are very similar, and the names are sometimes used interchangeably.

The PDE (26) can be derived using similar no-arbitrage arguments as in the Black-Scholes model. However, this time it is not possible to balance the hedging portfolio with only the underlying asset S_t. For instance, in [25] the PDE is derived by adding another option with a different expiration date into the hedging portfolio.

3.3 Jump-diffusion Models

The Black-Scholes and stochastic volatility models are examples of diffusion models, where the sample paths of the process are continuous. Stochastic volatility models are able to generate heavy tails in the return distribution for larger time intervals, which is a result of the accumulation of small moves over a sufficiently long time. However, large *sudden* changes under diffusion models are next to impossible[5]. The addition of jumps into the model generates heavy tails in returns for *short* time intervals, and allows large sudden changes in the underlying asset. This is particularly important from a risk management perspective, since the implication is that large losses are possible even in a short time interval. Similarly, as in the case of stochastic volatility, markets under jump-diffusion models are *incomplete*. The difference between diffusion and jump-diffusion models in risk management is well illustrated by the six standard deviation market movement example; see [13] for instance. Such (six sigma) movements are occasionally recorded in real world markets, but for a normal random variable an event that is six standard deviations from the mean has a probability less than 10^{-8}. Under the Black-Scholes model such a daily return would happen on average once in a million years!

A suitable stochastic process that allows jumps and diffusion is the Lévy process; see e.g. [13]. It has similar properties as the Brownian motion, but also allows discontinuities. Indeed, Brownian motion is a special case of the Lévy process. Another special case of the Lévy process is a pure jump process, where there is no diffusion component. Here, we are particularly interested in the jump-diffusion case, where both the diffusion and jump components are nonzero.

Lévy processes can be divided into finite and infinite (jump) activity cases. In the case of infinite activity, it is not necessary to introduce a Brownian component at all, since the jump activity is high enough to generate realistic small timescale behavior. This construction is important in terms of theoretical results. For example, small jumps with infinite activity drawn from a stable distribution start to resemble a diffusion process. After all, Brownian motion can be obtained as the limit of a random walk when time step sizes are taken to zero. Often in practical computations the infinite activity portion of the jump distribution is approximated with artificial diffusion, hence, the computation reduces to a finite

[5] Technically, it is possible to generate large sudden changes under a diffusion model by fine tuning the volatility to suddenly rise to extreme levels and then rapidly return close to normal levels. A jump would produce a similar effect naturally, however.

activity jump-diffusion case. This is one of the reasons why we choose to focus on finite activity jump-diffusion models.

The pioneering paper by Merton [53] proposed log-normally sized jumps that arrive according to a Poisson process. In general, the underlying asset S_t under jump-diffusion models follows the SDE

$$\frac{dS_t}{S_{t-}} = (\mu - \lambda\kappa)dt + \sigma dW(t) + (\eta - 1)dN_t, \tag{27}$$

where N_t is a Poisson process, κ is the expected relative jump size, λ is the rate of the Poisson process, and $\eta - 1$ is a random variable producing a jump from S_t to $S_t\eta$. In case of the Merton model, the random variable η is log-normally distributed. The expected relative jump size κ can be written $\kappa = \mathbb{E}(\eta - 1)$. If $\kappa \neq 0$, then the jump process $(\eta - 1)dN_t$ is *not* a martingale. This is the reason for the compensator term $-\lambda\kappa dt$ in the SDE (27). The notation S_{t-} in (27) means that the value before a (possible) jump is used, more precisely $S_{t-} := \lim_{u \to t} S_u$. By denoting $\gamma := \mu - \lambda\kappa$ and $\xi := \eta - 1$ the SDE (27) can be written more succinctly as

$$\frac{dS_t}{S_{t-}} = \gamma dt + \sigma dW(t) + \xi dN_t. \tag{28}$$

The following partial integro-differential equation (PIDE) can be derived for the evolution of the price $v := v(S_t, t)$ of a European option under jump-diffusion models

$$\frac{\partial v}{\partial t} + \frac{1}{2}\sigma^2 S_t^2 \frac{\partial^2 v}{\partial S_t^2} + (r - \lambda\kappa)S_t \frac{\partial v}{\partial S_t} - (r + \lambda)v + \lambda\left(\int_0^\infty v(yS_t, t)f(y)dy\right) = 0, \tag{29}$$

where $f(\cdot)$ is the density function of the jump size distribution. The PIDE (29) can be derived using standard no-arbitrage arguments; see [53, 71] for example. The formulation given above is valid only if the jump activity is finite. A more general formulation and derivation is given in [13], which employs the more general Lévy process framework.

3.4 Stochastic Volatility with Jumps

A natural extension of the models discussed previously is a combination of stochastic volatility and jumps. Bates proposed a model [9] with lognormal jumps in the asset price, and volatility modeled by the CIR process. This is essentially a combination of the Merton jump-diffusion model and the Heston stochastic volatility model. The aim is to have a complete model that accurately captures both the short and long term behavior of empirically observed financial time series. The volatility under stochastic volatility models is *persistent*, that is, changes in volatility have an impact on the future distribution of returns. On the other hand, jumps in the asset price are *transient*, i.e., the future behavior of the asset price process

is unaffected by a jump. Duffie et al. proposed arguably more realistic models with stochastic volatility and jumps in *both* the asset price and volatility [19], which leads to a volatility model that is both persistent and allows large sudden changes. The first model has jumps in the asset price and volatility that arrive independent of each other (SVIJ). The other model has jumps in the asset price and volatility that arrive concurrently (SVCJ). While both the SVIJ and SVCJ models allow large sudden changes in volatility, the SVCJ model in particular links the rapid changes in volatility with times of sudden shocks in the market. Eraker et al. further investigate the Bates, SVIJ, and SVCJ models, and they find that the SVIJ and SVCJ models produce the best fit to the market data [20]. The SVCJ model has the added advantage of easier parameter estimation since jumps in the asset price and volatility occur simultaneously.

The dynamics of the underlying asset S_t and variance V_t under the Bates model follow the SDEs

$$\begin{aligned} \frac{dS_t}{S_{t-}} &= \gamma dt + \sqrt{V_t} dW_t^S + \xi dN_t, \\ dV_t &= a(b - V_t)dt + \psi\sqrt{V_t} dW_t^V, \end{aligned} \tag{30}$$

where the jumps arrive at rate λ and the jump sizes ξ are drawn from a lognormal distribution, and the correlation between W_t^S and W_t^V is given as in (24). Similarly, for the SVIJ and SVCJ models, the dynamics of S_t and V_t are given by

$$\begin{aligned} \frac{dS_t}{S_{t-}} &= \gamma dt + \sqrt{V_{t-}} dW_t^S + \xi^S dN_t^S, \\ dV_t &= a(b - V_{t-})dt + \psi\sqrt{V_{t-}} dW_t^V + \xi^V dN_t^V, \end{aligned} \tag{31}$$

where jumps arrive at rates λ^S and λ^V and the sizes ξ^S,ξ^V are drawn from a lognormal and exponential distribution, respectively. In the case of the SVCJ model the jumps are concurrent $N_t^S = N_t^V$, and their sizes are correlated according to a jump correlation parameter ρ_J.

Finally, Table 2 summarizes the models formulated in this chapter. In addition, sample paths from several models are plotted in Figure 2 simulated with parameters listed in Table 3.

TABLE 2 Well-known financial market models

Type	Risky Asset Model	Volatility Model
Black-Scholes[10]	$\frac{dS_t}{S_t} = \mu dt + \sigma dW_t$	Constant
Jump-diffusion[53, 46]	$\frac{dS_t}{S_t} = \gamma dt + \sigma dW_t + \xi^S dN_t^S$	Constant
Stochastic Volatility (Heston)[30]	$\frac{dS_t}{S_t} = \mu dt + \sqrt{V_t} dW_t^S$	$dV_t = a(b - V_t)dt + \psi\sqrt{V_t} dW_t^V$
SV with Jumps in returns (Bates)[9]	$\frac{dS_t}{S_t} = \gamma dt + \sqrt{V_t} dW_t^S + \xi^S dN_t^S$	$dV_t = a(b - V_t)dt + \psi\sqrt{V_t} dW_t^V$
SV with Jumps in S and V (SVIJ) and (SVCJ)[19]	$\frac{dS_t}{S_t} = \gamma dt + \sqrt{V_t} dW_t^S + \xi^S dN_t^S$	$dV_t = a(b - V_t)dt + \psi\sqrt{V_t} dW_t^V + \xi^V dN_t^V$
CIR interest rate model[15]	$dr_t = a(b - r_t)dt + \sigma\sqrt{r_t} dW_t$	Constant
Hull-White/Vasicek interest rate models[37, 70]	$dr_t = a(b_t - c_t r_t)dt + \sigma dW_t$	Constant
S_t	risky asset price process	
V_t	(square-root) volatility process	
σ, ψ	volatility coefficient	
W_t, W_t^S, W_t^V	standard Brownian motion	
μ, γ	drift	
r_t	interest rate	
a	rate of mean reversion	
b	long term mean level	
$\xi^S dN_t^S$	Poisson process N_t^S with rate λ_S and jump size distribution ξ^S	

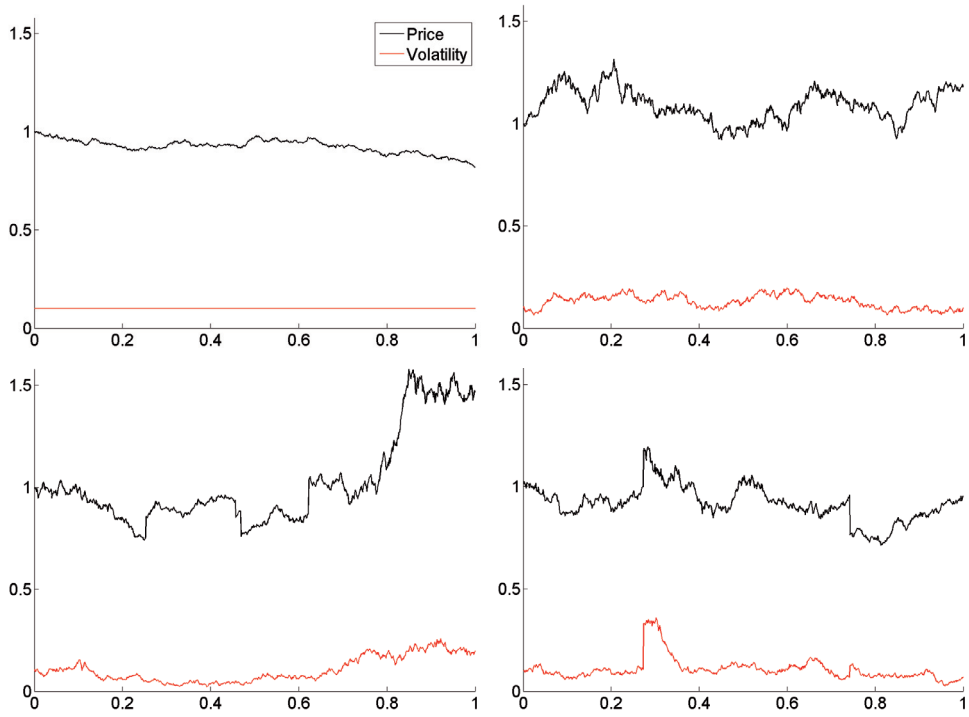

FIGURE 2 Sample paths from Black-Scholes (Top Left), Heston (Top Right), Bates (Bottom Left), and SVCJ (Bottom Right) simulated with parameters given in Table 3.

TABLE 3 Parameter setup of the sample path simulations

Expiry Time T	1
Drift rate μ	0.05
Rate of mean reversion a	5
Volatility (mean level) σ	0.1
Correlation coefficients ρ, ρ_J	0
Volatility of volatility ψ	0.5
Jump arrival rate λ	2

4 NUMERICAL SOLUTION METHODS

The Feynman-Kac formula establishes a link between certain stochastic processes and parabolic PDEs [45, 55, 56], particularly, PDEs that are of the form that were discussed in the previous chapter. As a consequence, the solution of the PDE can be written as a conditional expectation, and vice versa. Therefore, one can either solve the PDE directly, or alternatively evaluate the conditional expectation. In this chapter, we will consider numerical solution techniques for the PDE approach. We discretize the spatial dimension using the finite difference method. Alternative discretization methods include the finite element method and the binomial tree method.

Instead of solving the PDE numerically, one can evaluate the conditional expectation. In the special case of a European-style option under the Black-Scholes model, the conditional expectation has an analytic solution. For more general models the conditional expectation needs to be evaluated by Monte Carlo methods or numerical integration techniques. Monte Carlo is based on simulating a large number of sample paths for the underlying SDE, then the central limit theorem states that the option values generated by these paths converge in distribution to the conditional expectation. However, the convergence rate of Monte Carlo is very poor, and a more efficient approach, when available, is to employ semi-analytic numerical integration techniques, such as the Fourier transform based methods; see [21, 22, 51] for example.

The PDE formulation extends to American-style options with minimal effort, whereas the conditional expectation is directly valid only for European-style options. It is possible to extend Monte Carlo and numerical integration techniques to American-style options, but it requires a considerable amount of extra computation. For example, the standard approach to apply Monte Carlo methods to American-style options is to first simulate the corresponding paths under European-style exercise, and then go back and analyze each path separately to find the optimal exercise under American-style exercise [50]. Numerical integration techniques can also be extended for American-style options, but this generally requires the solution of a number of European option pricing problems, which yields an approximation of the continuous exercise possibility; see

e.g. [22]. As a general rule of thumb, analytic or semi-analytic solution methods are often available for European-style options, while American-style options typically require numerical solution.

4.1 Partial Integro-Differential Equation Formulation

In the previous chapter we formulated various PDEs and PIDEs that govern the price evolution of a European-style option contract. In practice, boundary conditions need to be introduced in order to have a completely defined problem. Consider the jump-diffusion PIDE in (29), for example. Let us invert the time dimension by setting $\tau = T - t$, so that we obtain a more typical initial value problem instead of a final value problem. Denote the underlying asset by $x := S_t$, and the partial derivatives by $v_\tau := \frac{\partial v}{\partial \tau}$, $v_x := \frac{\partial v}{\partial x}$, and $v_{xx} := \frac{\partial^2 v}{\partial x^2}$. Now, the PIDE reads

$$v_\tau = \frac{1}{2}\sigma^2 x^2 v_{xx} + (r - \lambda\kappa)xv_x - (r + \lambda)v + \lambda\left(\int_0^\infty v(xy, \tau)f(y)dy\right) =: Lv, \quad (32)$$

where we introduce Lv for future convenience. The PIDE holds for all $(x, \tau) \in [0, \infty) \times (0, T]$. The initial value of v at time $\tau = 0$ is known, and it is given by

$$v(x, 0) = g(x), \quad x \in [0, \infty), \quad (33)$$

where $g(x)$ is the payoff function of the option contract as described in Chapter 2. The boundary conditions for a European put option can be set as

$$\begin{aligned} v(0, \tau) &= Ke^{-r\tau}, \\ \lim_{x\to\infty} v(x, \tau) &= 0, \qquad \tau \in [0, T], \end{aligned} \quad (34)$$

where K is the strike price of the option, and r is the risk-free interest rate. Call options can be priced in a similar manner. Alternatively, the put-call parity formula can be employed in order to obtain the price of a call option, when the price of the corresponding put option is known. Complete formulations for the other models in Chapter 3 are similar, with the main difference of an extra spatial dimension in case of stochastic volatility.

4.2 American Option Early Exercise Constraint

The early exercise possibility of an American-style option can be handled in several ways. In the special case of the Black-Scholes model, the price of an American-style option can be obtained exactly as an infinite series [74]. Under a more general assumption, the so-called smooth pasting principle, which states that the price is smooth across the early exercise boundary, a free boundary problem can

be formulated for the price of an American option. Employing this approach, the front-fixing and front-tracking methods can be formulated. These methods have been studied in [54, 58, 72], and more recently in [31, 67, 75].

Taking a direct approach, the price of an American-style option satisfies a partial integro-differential *inequality*. The most common way to formulate this inequality is in the form of a linear complementarity problem (LCP). Zhang [73] described a variational inequality for the price of an American-style option under the Merton model. It can be shown to be equivalent to the LCP form [44]. Alternatively, these problems can be formulated with a Lagrange multiplier as in [42, 43].

The value v of an American-style option can be obtained as a solution to the following LCP

$$\begin{cases} (v_\tau - Lv) \geq 0, \quad v \geq g, \\ (v_\tau - Lv)(v - g) = 0, \end{cases} \tag{35}$$

where Lv is defined as in the PIDE (32). For an American put option the boundary conditions can be set as

$$\begin{aligned} v(0,\tau) &= K, \\ \lim_{x\to\infty} v(x,\tau) &= 0, \quad \tau \in [0,T]. \end{aligned} \tag{36}$$

In the special case where the early exercise region is simply connected to the boundary, which is the case for a simple put or call option, the LCP (35) can be solved by employing the Brennan-Schwartz algorithm [11]. This algorithm has optimal computational complexity and is very fast in practice. The projected successive over relaxation (PSOR) method [16, 33] can also be employed to obtain a solution to (35), however, it is not particularly efficient, especially under jump-diffusion models [PIV].

The penalty method is efficient and it can be employed to obtain an approximate solution to the LCP (35). Under the Black-Scholes model these methods were considered in [24, 54]. Under jump-diffusion models penalty methods were employed in [17, 66]. Another approach is to use the Lagrange multiplier formulation for the LCP, and employ an operator splitting method [39, 40], which results in a European option pricing problem at each time step and an update step. The operator splitting method produces an approximate solution to (35), where the error of the method was shown to be of the same order than the underlying time discretization scheme [40].

4.3 Space Discretization

To simplify notation, let us first divide the PIDE (32) into parts as follows

$$Lv = Dv + Jv, \tag{37}$$

where D is the differential operator and J is the integral operator defined by

$$\begin{aligned} Dv &= \frac{1}{2}\sigma^2 x^2 v_{xx} + (r - \lambda\kappa)xv_x - (r+\lambda)v, \\ Jv &= \lambda \left(\int_0^\infty v(\tau, xy) f(y) dy \right). \end{aligned} \tag{38}$$

The infinite space domain $[0, \infty)$ is truncated to $[0, X]$ with a sufficiently large X. Let us construct an N-node uniform grid

$$0 = x_1 < x_2 < \cdots < x_N = X. \tag{39}$$

Now, we can discretize the spatial derivatives in (37), which results in a semi-discrete system

$$v_\tau = \boldsymbol{D}\boldsymbol{v} + \boldsymbol{J}\boldsymbol{v} =: \boldsymbol{L}\boldsymbol{v}, \tag{40}$$

where $\boldsymbol{D}$ is a banded matrix, and $\boldsymbol{J}$ is a full matrix. For instance, the matrix $\boldsymbol{D}$ is *tridiagonal* under a standard finite difference discretization. See [63, PI] for more details on finite difference discretizations. Another popular approach is to adopt a Galerkin discretization in space and employ the finite element method; see [1, 2] for example. The matrix $\boldsymbol{J}$ results from the application of a suitable numerical quadrature on J. For American options, the space discretization leads to a semi-discrete LCP

$$\begin{cases} v_\tau - \boldsymbol{L}\boldsymbol{v} \geq \boldsymbol{0}, \quad \boldsymbol{v} \geq \boldsymbol{g}, \\ (v_\tau - \boldsymbol{L}\boldsymbol{v})^T (\boldsymbol{v} - \boldsymbol{g}) = 0, \end{cases} \tag{41}$$

where $\boldsymbol{g}$ is the discrete payoff function.

The integral term Jv of (38) can be evaluated numerically by employing standard numerical quadratures, such as the Simpson's rule or Gaussian quadrature. Alternatively, one can employ linear interpolation for v between spatial grid points, and then integrate this piecewise integral analytically; see [66, PI] for example. Either way, this results in a full matrix $\boldsymbol{J}$. Thus, adopting this direct approach leads to a full matrix-vector multiplication $\boldsymbol{J}\boldsymbol{v}$ requiring $O(N^2)$ operations. Later we will present some special techniques, that allow the computation of this multiplication in a more efficient manner.

4.4 Time Discretization

The numerical solution of the PDEs described in Chapter 3 is straightforward, since applying a standard space discretization on them leads to a banded (typically tridiagonal) matrix. The main difficulty is the numerical solution of models with jumps, which result in PIDEs, such as the PIDE (32). The presence of the (non-local) jump term causes the discretized matrix to be full.

Second-order accurate time discretization schemes are traditionally implicit schemes. However, employing a standard implicit time discretization scheme on such a system would require inverting a full matrix. This can be avoided by

employing iterative methods, for example, a fixed-point iteration for European options [65], or a similar iteration for American options [PI]. An efficient penalty iteration can also be employed on models with jumps [17].

Alternatively, the semi-discrete systems (40) and (41) can be discretized in time by adopting special implicit-explicit (IMEX) time discretization schemes. Typically, the jump term is treated explicitly, while the rest is handled implicitly. A first-order accurate IMEX-scheme was proposed in [14]. Second-order IMEX-schemes are usually two-step methods, such as the IMEX-midpoint scheme; see [48, 47] for an example in an option pricing context. The standard implicit schemes are *unconditionally* stable, whereas IMEX-schemes are *conditionally* stable.

In the following, we describe the iterative methods and IMEX-schemes in more detail. While we formulated the PIDE (32) under a one-dimensional jump-diffusion model, the semi-discrete systems (40) and (41) can be written in a similar form under more general models, such as the Bates and SVCJ models; see [PV].

4.4.1 Standard Implicit Schemes

In the special case of a European option, efficient Runge-Kutta style time discretization schemes are particularly efficient [12, 23]. However, these schemes include internal steps, which are computed by extrapolation. Therefore, this approach is not directly applicable to American options, since the extrapolation fails near the early exercise boundary. Here we adopt the more general second-order accurate Rannacher scheme [59], which consists of a few initial implicit Euler steps, followed by Crank-Nicolson steps. The implicit Euler steps are taken to damp the non-smooth initial condition (payoff function), since Crank-Nicolson is not L-stable and a non-smooth initial condition might cause spurious oscillations in the solution. The convergence of the Rannacher scheme was analyzed in [28].

Let us construct a $(M+1)$-node time grid with time step sizes $\Delta\tau = T/M$, except the first two steps under implicit Euler are taken with the step size $\Delta\tau/2$. This selection enables us to create the fully discretized matrix only once. Now, by applying the Rannacher time-stepping scheme onto the semi-discrete system (40) and using the θ-scheme notation we have

$$\frac{\boldsymbol{v}_{m+1} - \boldsymbol{v}_m}{\Delta\tau} = (\boldsymbol{D} + \boldsymbol{J})(\theta \boldsymbol{v}_{m+1} + (1-\theta)\boldsymbol{v}_m), \tag{42}$$

where the parameter $\theta := \theta_m$ is given by

$$\theta := \theta_m = \begin{cases} 1, & 1 \le m \le 2, \\ \dfrac{1}{2}, & m = 3, \ldots, M+1. \end{cases} \tag{43}$$

The Rannacher scheme is unconditionally stable, and if time steps are increased in such a way that the number implicit Euler steps is kept small, then the scheme is second-order convergent in time.

4.4.2 Implicit-Explicit Schemes

In the Rannacher scheme (42) above, we can see that it is necessary to invert the full matrix $\boldsymbol{D} + \boldsymbol{J}$ at each time step. The simple idea to avoid the inversion of a full matrix is to employ the IMEX-Euler scheme [13, 14], which reads

$$\frac{\boldsymbol{v}_{m+1} - \boldsymbol{v}_m}{\Delta\tau} = \boldsymbol{D}\boldsymbol{v}_{m+1} + \boldsymbol{J}\boldsymbol{v}_m. \tag{44}$$

Now it is only necessary to compute the full matrix-vector multiplication $\boldsymbol{J}\boldsymbol{v}$. However, the major drawback is that this scheme is only first-order accurate in time. Second-order accurate two-step IMEX schemes can be constructed by starting from an implicit scheme, and then combining it with an extrapolation procedure for the explicit part. These types of scheme were studied in [29, 38], and in an option pricing context in [47, 48, PIII].

The most commonly formulated two-step IMEX schemes include the IMEX-midpoint, IMEX-CNAB and IMEX-BDF2 schemes. Applied on the semi-discrete system (40), the IMEX-midpoint scheme reads

$$\frac{\boldsymbol{v}_{m+1} - \boldsymbol{v}_{m-1}}{2\Delta\tau} = \boldsymbol{J}\boldsymbol{v}_m + \boldsymbol{D}\left(\frac{\boldsymbol{v}_{m+1} + \boldsymbol{v}_{m-1}}{2}\right), \tag{45}$$

whereas the IMEX-CNAB (Crank-Nicolson, Adams-Bashforth) is given by

$$\frac{\boldsymbol{v}_{m+1} - \boldsymbol{v}_m}{\Delta\tau} = \boldsymbol{J}\left(\frac{3\boldsymbol{v}_m - \boldsymbol{v}_{m-1}}{2}\right) + \boldsymbol{D}\left(\frac{\boldsymbol{v}_{m+1} + \boldsymbol{v}_m}{2}\right), \tag{46}$$

and the IMEX-BDF2 scheme reads

$$\frac{\frac{3}{2}\boldsymbol{v}_{m+1} - 2\boldsymbol{v}_m + \frac{1}{2}\boldsymbol{v}_{m-1}}{\Delta\tau} = \boldsymbol{J}(2\boldsymbol{v}_m - \boldsymbol{v}_{m-1}) + \boldsymbol{D}\boldsymbol{v}_{m+1}. \tag{47}$$

These IMEX-schemes are *conditionally* stable; see [27, PIII] for stability analysis.

4.5 Iterative Methods

Employing standard time discretization schemes, such as the Rannacher scheme (42), on jump-diffusion models leads to systems with full matrices. In case of European options, these systems are linear systems of equations, whereas for American options they are systems of LCPs. In the following we describe some iterative methods that avoid the inversion of a full matrix when solving these problems.

4.5.1 Fixed-Point Iteration

Tavella and Randall proposed a stationary iterative method in [65] for European options, which reads

$$\boldsymbol{v}^{l+1} = \boldsymbol{T}^{-1}(\boldsymbol{b} + \theta\boldsymbol{J}\boldsymbol{v}^l), \quad l = 0, 1, \ldots, \tag{48}$$

where $\boldsymbol{T} = \boldsymbol{I}/\Delta\tau - \theta\boldsymbol{D}$, $\boldsymbol{b} = \boldsymbol{v}_m/\Delta\tau + (1-\theta)(\boldsymbol{D}+\boldsymbol{J})\boldsymbol{v}_m$, and $\boldsymbol{I}$ denotes the $N \times N$ identity matrix. Now by taking $\boldsymbol{v}_m$ as the initial guess, that is $\boldsymbol{v}^0 = \boldsymbol{v}_m$, and solving in sequence the linear systems (48), one obtains a convergent approximate solution to the original problem (42). Notice that $\boldsymbol{T}$ is a banded matrix, or in the case of a standard finite difference space discretization $\boldsymbol{T}$ is a tridiagonal matrix. Hence, (48) can be solved extremely efficiently by employing LU-factorization.

A similar iteration was proposed for American options in [PI]. First, let us denote the LCP

$$\boldsymbol{B}\boldsymbol{v} \geq \boldsymbol{b}, \quad \boldsymbol{v} \geq \boldsymbol{g}, \quad (\boldsymbol{B}\boldsymbol{v} - \boldsymbol{b})^T(\boldsymbol{v} - \boldsymbol{g}) = 0 \tag{49}$$

by LCP$(\boldsymbol{B}, \boldsymbol{v}, \boldsymbol{b}, \boldsymbol{g})$. Now the iterative method for American options reads

$$\text{LCP}(\boldsymbol{T}, \boldsymbol{v}^{l+1}, \theta\boldsymbol{J}\boldsymbol{v}^l + \boldsymbol{b}, \boldsymbol{g}), \quad l = 0, 1, \ldots. \tag{50}$$

Thus, the iteration leads to LCPs with banded matrices instead of full matrices. In [PI] it was shown that the iteration (50) converges at the rate

$$C = \frac{\theta\Delta\tau\lambda}{1 + \theta\Delta\tau(r+\lambda)}. \tag{51}$$

In typical circumstances $C \ll 1$, thus, the iteration converges rapidly.

4.5.2 Penalty Iteration

A penalty formulation was proposed by d'Halluin et al. in [17] to price American options under jump-diffusion models. The authors formulate the PIDE (32) as

$$v_\tau = Lv + \frac{1}{\epsilon}\max\{g - v, 0\}, \tag{52}$$

where the last term is a penalty term penalizing for the violation of the early exercise constraint, and ϵ is a small positive penalty parameter. By applying, for example, finite differences and the Rannacher scheme onto (52) we obtain nonlinear non-smooth systems of equations. Such systems can be solved by employing a semismooth Newton method; see [43, 76], for example. However, the semismooth Newton iteration would require solving linear systems with full matrices. Instead, we employ an approximate semismooth Newton method described in [17, 66]. Essentially, the jump term is dropped out of the coefficient matrix. Thus, we can iterate a sequence of tridiagonal systems given by

$$\boldsymbol{v}^{l+1} = \left(\boldsymbol{C}^l\right)^{-1}(\boldsymbol{b} + \theta\boldsymbol{J}\boldsymbol{v}^l + \boldsymbol{P}^l\boldsymbol{g}), \quad l = 0, 1, 2, \ldots, \tag{53}$$

where $\boldsymbol{C}^l$ and $\boldsymbol{b}$ are given by

$$\begin{aligned} \boldsymbol{C}^l &= \frac{1}{\Delta\tau}\boldsymbol{I} - \theta\boldsymbol{D} + \boldsymbol{P}^l \qquad \text{and} \\ \boldsymbol{b} &= \frac{1}{\Delta\tau}\boldsymbol{v}_m + (1-\theta)[\boldsymbol{D}+\boldsymbol{J}]\boldsymbol{v}_m. \end{aligned} \tag{54}$$

Above $\boldsymbol{P}^l$ is the diagonal penalty matrix enforcing the penalty term in (52). At each iteration $\boldsymbol{P}^l$ is updated, which is described in [PII]. The penalty iteration (53) has been shown [17] to rapidly converge to the unique solution of (52). The penalty method can be employed in combination with the IMEX-schemes as well [PII]. In this case the semismooth Newton method can be used directly, since the time discretization leads to a tridiagonal matrix.

4.6 Efficient Evaluation of the Jump Term

In the above described IMEX-schemes and iterative methods the inversion of a full matrix is avoided. A matrix-vector multiplication with the full matrix $\boldsymbol{J}$ is still required. The direct computation of such a multiplication would require $O(N^2)$ operations. A more efficient approach is to compute the multiplication as a convolution integral and then employ FFTs, which can be computed in $O(N \log N)$ operations. In the special case of the Kou model, an even more efficient special technique is available. The memoryless property of the density function f enables the use of efficient recursion formulas [66], which lead to the optimal computational complexity $O(N)$. Here we briefly describe these methodologies.

4.6.1 Convolution Form

The integral term

$$Jv = \lambda \left(\int_0^\infty v(\tau, xy) f(y) dy \right) \tag{55}$$

of (38) can be written as

$$I = \int_0^\infty v(\tau, xy) f(y) dy = \int_{-\infty}^\infty \bar{v}(\tau, \bar{x} + z) \bar{f}(z) dz, \tag{56}$$

by setting $\bar{x} = \log(x)$ and using the change of variable $z = \log(y)$. Above we denote $\bar{v}(\tau, z) = v(e^z)$ and $\bar{f}(z) = f(e^z)e^z$. The right-hand side of (56) is a convolution integral. Now if we have a uniform grid in $\bar{x}$ and we apply the trapezoidal rule on the convolution integral, then the resulting matrix is a Toeplitz matrix; see for example [PV]. This matrix can be embedded into a circulant matrix. A multiplication with a circulant matrix can be computed efficiently using FFTs by employing the convolution theorem.

A uniform grid in $\bar{x}$ is not very practical, since it is logarithmically uniform in the original variable x. Therefore, it is prudent to use interpolation to construct a uniform grid in $\bar{x}$ from an arbitrary grid in x. In the following we summarize the necessary steps to compute (56) by employing FFTs.

- Precompute and store the FFT of $\bar{f}(z)$ at points $z = \bar{x}_i$.
- Use $\boldsymbol{v}_m$ to interpolate from grid points in x to $\bar{x}$, which yields $\bar{\boldsymbol{v}}_m$.
- Take the FFT of $\bar{\boldsymbol{v}}_m$ and multiply it with the precomputed FFT.

– Take the the inverse FFT of the result of the previous step, and then interpolate it back to the grid points in x.

The FFTs require only $O(N \log N)$ operations as opposed to the $O(N^2)$ operations of the direct approach; see also [3, 18]. In case of the SVCJ model the FFTs need to be taken in two directions. In [PV] the formulations are given under the Bates and SVCJ models.

4.6.2 Recursion Formulas

In the special case of the Kou model with log-double-exponentially sized jumps, it is possible to compute the integral efficiently by employing recursion formulas as in [66]. First, by changing variables $y = z/x$ in (55) we can write

$$I = \int_0^\infty v(\tau, xy) f(y) dy = \int_0^\infty v(\tau, z) f(z/x)/x dz. \tag{57}$$

Recall the log-double-exponential distribution density function

$$f(y) = \begin{cases} q\alpha_2 y^{\alpha_2 - 1}, & y < 1 \\ p\alpha_1 y^{-\alpha_1 - 1}, & y \geq 1, \end{cases} \tag{58}$$

where $p, q, \alpha_1 > 1$, and α_2 are positive constants such that $p + q = 1$. We can substitute (58) into (57) and decompose the integral as $I = I^- + I^+$ to obtain

$$I^- = \int_0^x v(\tau, z) f(z/x)/x dz = q\alpha_2 x^{-\alpha_2} \int_0^x v(\tau, z) z^{\alpha_2 - 1} dz \tag{59}$$

and

$$I^+ = \int_x^\infty v(\tau, z) f(z/x)/x dz = p\alpha_1 x^{\alpha_1} \int_x^\infty v(\tau, z) z^{-\alpha_1 - 1} dz. \tag{60}$$

Now consider I^- at each grid point x_i

$$I_i^- = q\alpha_2 x_i^{-\alpha_2} \int_0^{x_i} v(\tau, z) z^{\alpha_2 - 1} dz, \tag{61}$$

which can be approximated by employing linear interpolation for v between grid points yielding a matrix A^- with elements $A^-_{i,j}$ given by

$$A^-_{i,j} = q\alpha_2 x_i^{-\alpha_2} \int_{x_j}^{x_{j+1}} \left(\frac{x_{j+1} - z}{\Delta x_j} v(\tau, x_j) + \frac{z - x_j}{\Delta x_j} v(\tau, x_{j+1}) \right) z^{\alpha_2 - 1} dz. \tag{62}$$

Due to the memoryless nature of the log-double-exponential distribution, it holds

$$A^-_{i+1,j} = A^-_{i,j} \left(\frac{x_{i+1}}{x_i} \right)^{-\alpha_2} \qquad j = 0, \ldots, i-1. \tag{63}$$

This leads to the recursion formula

$$I^-_{i+1} \approx I^-_i \left(\frac{x_i}{x_{i+1}} \right)^{\alpha_2} + A^-_{i+1,i} \quad \forall i = 1, \ldots, N-1. \tag{64}$$

Similarly for I^+ it is possible to derive the recursion formula

$$I^+_{i-1} \approx I^+_i \left(\frac{x_{i-1}}{x_i} \right)^{\alpha_1} + A^+_{i-1,i-1} \quad \forall i = 2, \ldots, N-1. \tag{65}$$

Therefore, we can compute $(\boldsymbol{Jv})_i = -\lambda(I^-_i + I^+_i)$ for $i = 1, \ldots, N-1$ resulting in the computational complexity $O(N)$. In [66] it is shown that this approximation is second-order accurate.

5 CONTRIBUTION OF INCLUDED ARTICLES

This chapter provides a summary of each of the five included articles and presents their main contributions. The author's personal contribution to each of these articles is pointed out.

5.1 Article I

This article presents an iterative method for pricing American options under jump-diffusion models. The convergence of this method is proved in a general LCP setting, and then the rate of convergence of the method is given under a typical discretization. The main result of the article, Theorem 1, was commented in the peer review process to be a very nice result and useful in practice. The proposed iterative method was employed and analyzed further in [5, 34]. In [34] the iterative method is generalized for regime switching models, and in [5] the method is shown superior in terms of convergence bounds and numerical results compared to the other considered method.

The author did the majority of the writing, wrote the computer program and performed the numerical experiments. The co-author came up with the idea for the iterative method, and wrote the mathematical convergence proof. The author received an award from the IMACS association for this article.

5.2 Article II

A survey of finite difference methods for pricing American options under jump-diffusion models is given in this article. In addition, a comparison between six different approaches is presented. The article compares various ways of treating the early exercise constraint of the American option pricing problem. These include the Brennan-Schwartz algorithm, operator splitting method and penalty

method. These methods are then combined with either a traditional time discretization method combined with an iteration, or the IMEX-midpoint scheme. The resulting combinations are then compared in numerical experiments. The numerical experiments show that the iterative methods cannot compete in performance with the IMEX-midpoint scheme. In the peer review process the article was commented to be valuable due to bringing clarity to the large pool of papers published on this topic.

The author wrote the majority of the article and performed the numerical experiments. The co-author wrote the introduction and the survey sections.

5.3 Article III

A family of IMEX time discretization schemes for pricing options under jump-diffusion models is proposed in this article. The stability of the proposed methods is analyzed with Fourier stability analysis. It is found that the stability properties of the IMEX-midpoint scheme are inadequate in some circumstances. On the other hand, the IMEX-BDF2 and the IMEX-CNAB schemes show favorable stability properties. Furthermore, the IMEX-CNAB scheme produced the smallest error in numerical experiments, which leads us to recommend this time discretization method for the purposes of option pricing under jump-diffusion models.

The author wrote the majority of the article and performed the numerical experiments. The mathematical stability proofs were written in collaboration with the author and the co-author.

5.4 Article IV

In this article the iterative method proposed in [PI] is applied to price American options under the Bates model. At each iteration an LCP needs to be solved. These LCPs are solved using either the projected successive overrelaxation (PSOR) method [16] or the projected algebraic multigrid (PAMG) method [68]. While the accuracy of the considered methods is similar, the iterative method combined with the PAMG method turns out to be an order of magnitude faster when the discretizations are refined.

The co-authors wrote the majority of the article, and performed the numerical experiments. The author wrote several miscellaneous parts, especially related to the iterative method, and was involved in revising the article.

5.5 Article V

The IMEX-CNAB scheme, promoted in article [PIII], is employed in this article to price European and American options under the Bates and the SVCJ models. To the best of our knowledge, this is the first article dealing with the numerical pricing of American options under the SVCJ model. Due to the application of the IMEX-CNAB scheme, the resulting matrices are block tridiagonal. Thus, they can be solved by employing LU decomposition. Alternatively, they can be iterated under an algebraic multigrid (AMG) method [61]. The necessary matrix-vector multiplications are computed efficiently using FFTs. This procedure is described in detail for the Bates and the SVCJ models. The operator splitting (OS) method [39, 40] is employed to enable the use of LU decomposition for American options as well. Similarly, the PAMG method [68] is employed instead of the AMG for American options. The LU+OS methodology turns out to be surprisingly fast and accurate.

The author wrote the majority of the abstract, introduction, and the numerical experiments sections. The co-authors wrote the computer program and the method descriptions.

6 CONCLUSIONS

Option contracts play a fundamental role in the financial industry. They are used in purposes ranging from leveraging speculative bets to protecting against risk in investments. Only recently in 1973, a rigorously justified answer to the question of the fair price of an option contract was given by Black, Scholes, and Merton. Over the years, however, it has become evident that the Black-Scholes model is in many ways a crude simplification of real-world market behavior. The proposed corrections to the Black-Scholes model often fall into two major categories: stochastic volatility and jumps.

It can be argued that many of the problems with the Black-Scholes model can be fixed by introducing stochastic volatility. There is one feature that these diffusive models lack, however, which is the possibility of large sudden changes. It is well known that occasionally there is a large sudden change in the value of an asset. This might be due to quarterly reports, market distress, a technical breakthrough, or even an unforeseen disaster. Whatever the case, it remains that the impact of such movements is large even if the probability is small. For option contracts this is especially important from a risk management perspective. After all, the implication is that large losses are possible even in a small time window.

In this dissertation we focus on option pricing models with jumps. Some models combine jumps with stochastic volatility, such as the Bates and the SVCJ models. Others simply augment the Black-Scholes model with jumps, e.g., the Merton and Kou jump-diffusion models. Perhaps one of the reasons why these models are not in widespread use is that the solution to such option pricing models is more difficult to obtain.

Here we study numerical methods that solve these models employing PIDE formulations. At least second-order accurate time discretization methods are needed to obtain accurate option prices. Due to jumps, standard time discretization methods would require the inversion of a full matrix. A more efficient approach is to employ iterative methods, which iterate a sequence of systems with banded coefficient matrices. Alternatively, special IMEX time discretization methods can be employed to avoid the inversion of a full matrix. In both cases full matrix-vector multiplications are still necessary. These multiplications can be

computed efficiently by employing FFTs or recursion formulas. The application of these methodologies results in algorithms capable of pricing options under jump-diffusion processes in milliseconds, or a few seconds at most.

YHTEENVETO (FINNISH SUMMARY)

Vuonna 1973 Black ja Scholes julkaisivat uraauurtavan artikkelin optiohinnoittelusta. Black-Scholes mallissa kohde-etuutta (esim. osake) mallinnetaan satunnaisen Brownin liikkeen avulla. Ehkä suurimmat epäkohdat mallissa ovat oletukset vakiosuuruisesta markkinoiden volatiliteetista ja polkujen jatkuvuudesta. Viime vuosina erityisesti stokastisen volatiliteetin mallit ovat levinneet laajaan käyttöön. Voidaankin sanoa, että volatiliteetin muuttaminen stokastiseksi suureeksi ratkaisee monia ongelmia alkuperäisestä Black-Scholes mallista.

Yllä kuvattuja malleja voidaan kutsua diffuusiomalleiksi. Niissä tapahtuvat muutokset ovat asteittaisia. Suuret muutokset diffuusiomallien alaisuudessa vaativat aina enemmän tai vähemmän aikaa. Tiedetään kuitenkin, että markkinoilla tapahtuu ajoittain erittäin nopeita muutoksia. Tämänlaisten muutoksien mallintaminen diffuusiomallien avulla on erittäin vaikeaa ellei mahdotonta. Onkin perusteltua, että malliin tulisi lisätä mahdollisuus isoihin hyppyihin. Tällöin puhutaan hyppydiffuusiomalleista. Erityisesti lyhytikäisten optioiden hinnoittelussa on ratkaiseva ero sillä, mikä on hinnoittelumallin mukaan lyhyessä ajassa mahdollista. Vaikka hypyn todennäköisyys on erittäin pieni, niin sen vaikutus option hintaan on merkittävä. Erityisesti riskienhallinnan kannalta on kriittistä tiedostaa, että suuret tappiot ovat mahdollisia myös erittäin pienessä ajassa.

Yksi tapa hinnoitella optio on ratkaista osittaisdifferentiaaliyhtälö (ODY). Hyppydiffuusiomallien tapauksessa kyseessä on tarkemmin ottaen osittaisintegrodifferentiaaliyhtälö, koska hypyt lisäävät yhtälöön integraalin. Tämän tutkimuksen kohteena ovat erityisesti numeeriset menetelmät tämänlaisten yhtälöiden tehokasta ratkaisua varten. Työ on nimeltään *Numeerisia menetelmiä optiohinnoitteluun hyppydiffuusioprosessien alaisuudessa*. Ehkä yksi syy miksi hyppydiffuusiomallien käyttö on vielä varsin harvinaista on se, että optiohinnoittelu niiden alaisuudessa on selvästi vaikeampaa.

Ongelman diskretisointi johtaa tyypillisesti lineaariseen yhtälöryhmään, joka voidaan esittää matriisina. Hypyistä johtuva epälokaali integraali muuttaa tämän matriisin täydeksi. Perinteiset toisen kertaluokan aikadiskretisointimenetelmät vaativat tämän matriisin kääntämisen. Koska matriisi on tässä tapauksessa täysi, niin kääntämisen sijaan on paljon tehokkaampaa ratkaista ongelma iteratiivisilla menetelmillä, jotka antavat tarkan approksimaation ratkaisusta kääntämällä sarjan nauhamatriiseja. Vaihtoehtoisesti voidaan käyttää erikoisia implisiittieksplisiitti (IMEX) aikadiskretisointimenetelmiä, joiden avulla tehtävä muodostuu suoraan nauhamaiseksi. Molemmissa tapauksissa on silti tarpeellista suorittaa matriisi kertaa vektori kertolasku täydellä matriisilla. Tämä kertolasku voidaan suorittaa tehokkaasti käyttämällä Fourier-muunnoksia tai rekursiokaavoja. Kuvattujen menetelmien avulla on mahdollista kehittää algoritmeja, jotka pystyvät hinnoittelemaan optioita hyppydiffuusioprosessien alaisuudessa millisekunneissa, tai pahimmillaan muutamassa sekunnissa.

REFERENCES

[1] Y. ACHDOU AND O. PIRONNEAU, *Computational methods for option pricing*, vol. 30 of Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005.

[2] A. ALMENDRAL AND C. W. OOSTERLEE, *Numerical valuation of options with jumps in the underlying*, Applied Numerical Mathematics. An IMACS Journal, 53 (2005), pp. 1–18.

[3] L. ANDERSEN AND J. ANDREASEN, *Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing*, Review of Derivatives Research, 4 (2000), pp. 231–262.

[4] T. G. ANDERSEN, L. BENZONI, AND J. LUND, *An empirical investigation of continuous-time equity return models*, The Journal of Finance, 57 (2002), pp. 1239–1284.

[5] J. BABBIN, P. FORSYTH, AND G. LABAHN, *A comparison of iterated optimal stopping and local policy iteration for american options under regime switching*, Journal of Scientific Computing, (2011), pp. 1–22.

[6] L. BACHELIER, *Théorie de la spéculation*, Gauthier-Villars, 1900.

[7] C. A. BALL AND A. ROMA, *Stochastic volatility option pricing*, Journal of Financial and Quantitative Analysis, 29 (1994).

[8] E. BARKER ET AL., *The politics of Aristotle*, Oxford University Press, 1963.

[9] D. S. BATES, *Jumps and stochastic volatility: Exchange rate processes implicit Deutsche mark options*, Review of Financial Studies, 9 (1996), pp. 69–107.

[10] F. BLACK AND M. SCHOLES, *The pricing of options and corporate liabilities*, Journal of Political Economy, 81 (1973), pp. 637–654.

[11] M. J. BRENNAN AND E. S. SCHWARTZ, *The valuation of American put options*, Journal of Finance, 32 (1977), pp. 449–462.

[12] M. BRIANI, R. NATALINI, AND G. RUSSO, *Implicit-explicit numerical schemes for jump-diffusion processes*, Calcolo. A Quarterly on Numerical Analysis and Theory of Computation, 44 (2007), pp. 33–57.

[13] R. CONT AND P. TANKOV, *Financial modelling with jump processes*, Chapman & Hall/CRC, Boca Raton, FL, 2004.

[14] R. CONT AND E. VOLTCHKOVA, *A finite difference scheme for option pricing in jump diffusion and exponential Lévy models*, SIAM Journal on Numerical Analysis, 43 (2005), pp. 1596–1626.

[15] J. Cox, J. Ingersoll, and S. Ross, *A theory of the term structure of interest rates*, Econometrica: Journal of the Econometric Society, (1985), pp. 385–407.

[16] C. W. Cryer, *The solution of a quadratic programming problem using systematic overrelaxation*, SIAM Journal on Control and Optimization, 9 (1971), pp. 385–392.

[17] Y. d'Halluin, P. A. Forsyth, and G. Labahn, *A penalty method for American options with jump diffusion processes*, Numerische Mathematik, 97 (2004), pp. 321–352.

[18] Y. d'Halluin, P. A. Forsyth, and K. R. Vetzal, *Robust numerical methods for contingent claims under jump diffusion processes*, IMA Journal of Numerical Analysis, 25 (2005), pp. 87–112.

[19] D. Duffie, J. Pan, and K. Singleton, *Transform analysis and asset pricing for affine jump-diffusions*, Econometrica, 68 (2000), pp. 1343–1376.

[20] B. Eraker, M. Johannes, and N. Polson, *The impact of jumps in volatility and returns*, J. Finance, 58 (2003), pp. 1269–1300.

[21] F. Fang and C. W. Oosterlee, *A novel pricing method for European options based on Fourier-cosine series expansions*, SIAM Journal on Scientific Computing, 31 (2008/09), pp. 826–848.

[22] ——, *Pricing early-exercise and discrete barrier options by Fourier-cosine series expansions*, Numerische Mathematik, 114 (2009), pp. 27–62.

[23] L. Feng and V. Linetsky, *Pricing options in jump-diffusion models: an extrapolation approach*, Operations Research, 56 (2008), pp. 304–325.

[24] P. A. Forsyth and K. R. Vetzal, *Quadratic convergence for valuing American options using a penalty method*, SIAM Journal on Scientific Computing, 23 (2002), pp. 2095–2122.

[25] J. Fouque, G. Papanicolaou, and K. Sircar, *Derivatives in financial markets with stochastic volatility*, Cambridge University Press, 2000.

[26] ——, *Mean-reverting stochastic volatility*, International Journal of Theoretical and Applied Finance, 3 (2000), pp. 101–142.

[27] J. Frank, W. Hundsdorfer, and J. Verwer, *On the stability of implicit-explicit linear multistep methods*, Applied Numerical Mathematics. An IMACS Journal, 25 (1997), pp. 193–205.

[28] M. B. Giles and R. Carter, *Convergence analysis of Crank-Nicolson and Rannacher time-marching*, Journal of Computational Finance, 9 (2006), pp. 89–112.

[29] E. Hairer, S. Norsett, and G. Wanner, *Solving Ordinary Differential Equations. I: Nonstiff Problems*, Springer, 1993.

[30] S. L. HESTON, *A closed-form solution for options with stochastic volatility with applications to bond and currency options*, Review of Financial Studies, 6 (1993), pp. 327–343.

[31] A. D. HOLMES AND H. YANG, *A front-fixing finite element method for the valuation of American options*, SIAM Journal on Scientific Computing, 30 (2008), pp. 2158–2180.

[32] C. A. HOLT AND S. K. LAURY, *Risk aversion and incentive effects*, American Economic Review, 92 (2002), pp. 1644–1655.

[33] J. HUANG AND J.-S. PANG, *Option pricing and linear complementarity*, Journal of Computational Finance, 2 (1998), pp. 31–60.

[34] Y. HUANG, P. FORSYTH, AND G. LABAHN, *Inexact arithmetic considerations for direct control and penalty methods: American options under jump diffusion*, Applied Numerical Mathematics. An IMACS Journal, (2013).

[35] J. HULL, *Options, futures and other derivatives, 8th edit*, 2011.

[36] J. HULL AND A. WHITE, *The pricing of options on assets with stochastic volatilities*, The Journal of Finance, 42 (1987), pp. 281–300.

[37] J. HULL AND A. WHITE, *Pricing interest-rate-derivative securities*, Review of Financial Studies, 3 (1990), pp. 573–592.

[38] W. HUNDSDORFER AND J. VERWER, *A note on splitting errors for advection-reaction equations*, Applied Numerical Mathematics. An IMACS Journal, 18 (1995), pp. 191–199.

[39] S. IKONEN AND J. TOIVANEN, *Operator splitting methods for American option pricing*, Applied Mathematics Letters. An International Journal of Rapid Publication, 17 (2004), pp. 809–814.

[40] ——, *Operator splitting methods for pricing American options under stochastic volatility*, Numerische Mathematik, 113 (2009), pp. 299–324.

[41] K. ITÔ, *Stochastic integral*, Proceedings of the Japan Academy, Series A, Mathematical Sciences, 20 (1944), pp. 519–524.

[42] K. ITO AND K. KUNISCH, *Parabolic variational inequalities: the Lagrange multiplier approach*, Journal de Mathématiques Pures et Appliquées. Neuvième Série, 85 (2006), pp. 415–449.

[43] K. ITO AND J. TOIVANEN, *Lagrange multiplier approach with optimized finite difference stencils for pricing American options under stochastic volatility*, SIAM Journal on Scientific Computing, 31 (2009), pp. 2646–2664.

[44] P. JAILLET, D. LAMBERTON, AND B. LAPEYRE, *Variational inequalities and the pricing of American options*, Acta Applicandae Mathematicae. An International Survey Journal on Applying Mathematics and Mathematical Applications, 21 (1990), pp. 263–289.

[45] M. KAC, *On distributions of certain wiener functionals*, Transactions of the American Mathematical Society, 65 (1949), pp. 1–13.

[46] S. G. KOU, *A jump-diffusion model for option pricing*, Management Science, 48 (2002), pp. 1086–1101.

[47] Y. KWON AND Y. LEE, *A second-order finite difference method for option pricing under jump-diffusion models*, SIAM Journal on Numerical Analysis, 49 (2011), pp. 2598–2617.

[48] ——, *A second-order tridiagonal method for American options under jump-diffusion models*, SIAM Journal on Scientific Computing, 33 (2011), pp. 1860–1872.

[49] C. G. LAMOUREUX AND W. D. LASTRAPES, *Forecasting stock-return variance: Toward an understanding of stochastic implied volatilities*, Review of Financial Studies, 6 (1993), pp. 293–326.

[50] F. A. LONGSTAFF AND E. S. SCHWARTZ, *Valuing american options by simulation: A simple least-squares approach*, Review of Financial studies, 14 (2001), pp. 113–147.

[51] R. LORD, F. FANG, F. BERVOETS, AND C. W. OOSTERLEE, *A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes*, SIAM Journal on Scientific Computing, 30 (2008), pp. 1678–1705.

[52] R. C. MERTON, *Theory of rational option pricing*, The Bell Journal of Economics and Management Science, 4 (1973), pp. 141–183.

[53] ——, *Option pricing when underlying stock returns are discontinuous*, Journal of Financial Economics, 3 (1976), pp. 125–144.

[54] B. F. NIELSEN, O. SKAVHAUG, AND A. TVEITO, *Penalty and front-fixing methods for the numerical solution of American option problems*, Journal of Computational Finance, 5 (2002), pp. 69–97.

[55] D. NUALART AND W. SCHOUTENS, *Backward stochastic differential equations and feynman-kac formula for lévy processes, with applications in finance*, Bernoulli, 7 (2001), pp. 761–776.

[56] B. ØKSENDAL, *Stochastic differential equations*, Springer, 2003.

[57] J. PAN, *The jump-risk premia implicit in options: Evidence from an integrated time-series study*, Journal of Financial Economics, 63 (2002), pp. 3–50.

[58] K. N. PANTAZOPOULOS, E. N. HOUSTIS, AND S. KORTESIS, *Front-tracking finite difference methods for the valuation of American options*, Computational Economics, 12 (1998), pp. 255–273.

[59] R. RANNACHER, *Finite element solution of diffusion problems with irregular data*, Numerische Mathematik, 43 (1984), pp. 309–327.

[60] R. REBONATO, *Volatility and Correlation in the Pricing of Equity, FX, and Interest-rate Options*, John Wiley, 1999.

[61] J. W. RUGE AND K. STÜBEN, *Algebraic multigrid*, in Multigrid methods, vol. 3 of Frontiers Appl. Math., SIAM, Philadelphia, PA, 1987, pp. 73–130.

[62] L. O. SCOTT, *Option pricing when the variance changes randomly: Theory, estimation, and an application*, Journal of Financial and Quantitative analysis, 22 (1987), pp. 419–438.

[63] R. SEYDEL, *Tools for computational finance*, Springer, 2012.

[64] E. M. STEIN AND J. C. STEIN, *Stock price distributions with stochastic volatility: an analytic approach*, Review of Financial Studies, 4 (1991), pp. 727–752.

[65] D. TAVELLA AND C. RANDALL, *Pricing financial instruments: The finite difference method*, John Wiley & Sons, 2000.

[66] J. TOIVANEN, *Numerical valuation of European and American options under Kou's jump-diffusion model*, SIAM Journal on Scientific Computing, 30 (2008), pp. 1949–1970.

[67] ——, *A high-order front-tracking finite difference method for pricing American options under jump-diffusion models*, Journal of Computational Finance, 13 (2010), pp. 61–79.

[68] J. TOIVANEN AND C. W. OOSTERLEE, *A projected algebraic multigrid method for linear complementarity problems*, Numerical Mathematics. Theory, Methods and Applications, 5 (2012), pp. 85–98.

[69] G. E. UHLENBECK AND L. S. ORNSTEIN, *On the theory of the Brownian motion*, Physical review, 36 (1930), p. 823.

[70] O. VASICEK, *An equilibrium characterization of the term structure*, Journal of Financial Economics, 5 (1977), pp. 177–188.

[71] P. WILMOTT, *Derivatives*, John Wiley & Sons Ltd., Chichester, 1998.

[72] L. WU AND Y. K. KWOK, *A front-fixing finite difference method for the valuation of American options*, Journal of Financial Engineering, 6 (1997), pp. 83–97.

[73] X. L. ZHANG, *Numerical analysis of American option pricing in a jump-diffusion model*, Mathematics of Operations Research, 22 (1997), pp. 668–690.

[74] S.-P. ZHU, *An exact and explicit solution for the valuation of American put options*, Quantitative Finance, 6 (2006), pp. 229–242.

[75] S.-P. ZHU AND J. ZHANG, *A new predictor-corrector scheme for valuing American puts*, Applied Mathematics and Computation, 217 (2011), pp. 4439–4452.

[76] R. ZVAN, P. A. FORSYTH, AND K. R. VETZAL, *Penalty methods for American options with stochastic volatility*, Journal of Computational and Applied Mathematics, 91 (1998), pp. 199–218.

ORIGINAL PAPERS

PI

AN ITERATIVE METHOD FOR PRICING AMERICAN OPTIONS UNDER JUMP-DIFFUSION MODELS

by

S. Salmi and J. Toivanen 2011

Applied Numerical Mathematics, Volume 61, Issue 7, pp. 821-831

Applied Numerical Mathematics 61 (2011) 821–831

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

An iterative method for pricing American options under jump-diffusion models

Santtu Salmi [a,*], Jari Toivanen [b,a]

[a] Department of Mathematical Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland
[b] Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305, USA

ARTICLE INFO

Article history:
Received 22 June 2010
Received in revised form 7 January 2011
Accepted 7 February 2011
Available online 18 February 2011

Keywords:
American option
Jump-diffusion model
Finite difference method
Linear complementarity problem
Iterative method

ABSTRACT

We propose an iterative method for pricing American options under jump-diffusion models. A finite difference discretization is performed on the partial integro-differential equation, and the American option pricing problem is formulated as a linear complementarity problem (LCP). Jump-diffusion models include an integral term, which causes the resulting system to be dense. We propose an iteration to solve the LCPs efficiently and prove its convergence. Numerical examples with Kou's and Merton's jump-diffusion models show that the resulting iteration converges rapidly.

1. Introduction

It is widely recognized that the classic option pricing model proposed in 1973 by Black and Scholes in [5] and Merton in [19], does not ideally fit observed empirical market data. Two identified empirical features have been under much attention: (1) skewed distribution with higher peak and heavier tails (i.e. leptokurtic behavior) of the return distribution and (2) the volatility smile [3].

Many studies have been undergone to propose modifications to the Black–Scholes model to explain these phenomena. Here we focus on jump-diffusion models proposed by Kou in [17], and by Merton in [20]. These models have finite jump activity, unlike the more general approach with possibly infinite jump activity proposed by Carr, Geman, Madan and Yor in [7]. Another approach is to consider stochastic volatility models with jumps. The model proposed by Bates in [4] with jumps only in the value of the underlying asset is an example of such an approach. More general jump-diffusion models with stochastic volatility are considered in [10], for example.

Jumps can have a large impact on the price of an option, especially when near expiry. Merton's jump-diffusion model with the log-normal distribution is better able to produce the volatility smile phenomenon. However, the use of the log-double-exponential distribution instead of the log-normal distribution allows the introduction of asymmetric leptokurtic features. Kou argues in [17] that this model better matches empirical data without adding much complexity to the model. The downside of moving to the log-double-exponential distribution is that it uses more parameters than the log-normal distribution. Moreover, many theoretical results are only valid for the log-normal distribution [26].

A solution to a jump-diffusion model can be obtained by solving a partial integro-differential equation (PIDE). Due to the integral term the discretization leads to a full matrix. Direct solution methods are usually too expensive with a full

* Corresponding author. Tel.: +358 40 5818687; fax: +358 14 2602771.
E-mail addresses: ssalmi@jyu.fi (S. Salmi), toivanen@stanford.edu (J. Toivanen).

0168-9274/$30.00
doi:10.1016/j.apnum.2011.02.002

matrix, and therefore other numerical methods should be considered. Several methods have been proposed to approximate the linear complementarity problems resulting from American option pricing. These include a penalty method presented by d'Halluin, Forsyth and Labahn in [8] and an operator splitting method presented by Ikonen and Toivanen in [14,16]. One alternative approach to PIDEs is to employ a risk-neutral valuation formula and evaluate it quickly using FFT (Fast Fourier Transform) [11,18].

Tavella and Randall in [23] described a stationary iterative method for pricing European options. Here, we propose a generalization of this iterative method to price American options. Using this approach, each iteration requires the solution of LCP with a banded or sparse matrix instead of the full matrix. Under Kou's and Merton's models the banded matrix is tridiagonal. Brennan and Schwartz algorithm [6] can be used to solve these LCPs. While we do not consider any multi-factor jump-diffusion models, for example, stochastic volatility models with jumps, the iteration is also applicable for such models.

The outline of this paper is the following. In Section 2, Merton's and Kou's jump-diffusion models are introduced. In Section 3, a finite difference discretization for these models is presented. Section 4 describes the iterative methods used to solve the resulting systems of linear equations and linear complementarity problems for European and American options, respectively. Also, the Brennan and Schwartz algorithm is briefly introduced. Numerical experiments are given in Section 5, and finally Section 6 contains conclusions. The main contributions of this paper are the iterative method proposed in Section 4.1, and numerical experiments in Section 5.

2. Models for European and American options

The value of the underlying asset x under the classic Black–Scholes model [5] is given by

$$\frac{dx(t)}{x(t-)} = \mu\,dt + \sigma\,dW(t), \tag{1}$$

where μ is the drift rate, σ is the volatility and $W(t)$ is a standard Brownian motion. In general, a finite activity jump term is introduced into the model as follows

$$\frac{dx(t)}{x(t-)} = \mu\,dt + \sigma\,dW(t) + d\left(\sum_{j=1}^{N(t)} V_j\right), \tag{2}$$

where $N(t)$ is a Poisson process with rate λ and the set $\{V_j\}$ is a sequence of independent identically distributed random variables. Under Merton's jump-diffusion model the set $\{V_j\}$ is from the log-normal distribution with the density

$$f_{ln}(y) := \frac{1}{y\delta\sqrt{2\pi}} \exp\left(-\frac{(\log y - \gamma)^2}{2\delta^2}\right), \tag{3}$$

whereas under Kou's jump-diffusion model the set $\{V_j\}$ is from a distribution with the log-double-exponential density

$$f_{ld}(y) := \begin{cases} q\alpha_2 y^{\alpha_2 - 1}, & y < 1, \\ p\alpha_1 y^{-\alpha_1 - 1}, & y \geqslant 1, \end{cases} \tag{4}$$

where $p, q, \alpha_1 > 1$, and α_2 are positive constants such that $p + q = 1$.

Under the assumptions of the general jump-diffusion model (2), the value v of a European option can be obtained by solving a final value problem defined by a backward PIDE

$$v_t = Lv = -\frac{1}{2}\sigma^2 x^2 v_{xx} + (r - \lambda\kappa)xv_x + (r + \lambda)v - \lambda\left(\int_0^\infty v(t, xy) f(y)\,dy\right), \tag{5}$$

for all $(t, x) \in [0, T) \times [0, \infty)$. Above r is the (continuously compounded) risk free interest rate, f is the density function, and κ is the expected relative jump size. The final value of v is given by

$$v(T, x) = g(x), \quad x \in \mathbb{R}_+, \tag{6}$$

where $g(x)$ is the payoff function of the option contract. For a put option, it is

$$g(x) = \max\{K - x, 0\}, \tag{7}$$

where K is the strike price. The boundary conditions for a European put option are given by

$$\begin{aligned} &v(t, 0) = Ke^{-r(T-t)}, \\ &\lim_{x\to\infty} v(t, x) = 0, \quad t \in [0, T]. \end{aligned} \tag{8}$$

American options can be exercised at any time before expiry. Due to this, an additional constraint has to be introduced to the model to avoid arbitrage opportunities. The value v of an American option can be obtained by solving an LCP

$$\begin{cases} (v_t - Lv) \geqslant 0, \quad v \geqslant g, \\ (v_t - Lv)(v - g) = 0. \end{cases} \tag{9}$$

For American put options the behavior on the boundaries is given by

$$\begin{aligned} &v(t,0) = K, \\ &\lim_{x\to\infty} v(t,x) = 0, \quad t \in [0,T]; \end{aligned} \tag{10}$$

see [13] for more information.

3. Discretization

3.1. Discretization of spatial derivatives

We use finite differences to obtain an approximate solution. The infinite space domain is truncated to $[0, X]$ with a sufficiently large X to avoid an unacceptably large truncation error. The value of v at X is set to be $g(x)$. An n nodes grid

$$0 = x_1 < x_2 < \cdots < x_n = X \tag{11}$$

is used. The space derivatives of Eq. (5) are approximated with central-differences

$$v_x(t,x_i) \approx \frac{v_{i+1}(t) - v_{i-1}(t)}{\Delta x_{i-1} + \Delta x_i} \tag{12}$$

and

$$v_{xx}(t,x_i) \approx \frac{2[\Delta x_{i-1} v_{i+1}(t) - (\Delta x_{i-1} + \Delta x_i) v_i(t) + \Delta x_i v_{i-1}(t)]}{\Delta x_{i-1}\Delta x_i(\Delta x_{i-1} + \Delta x_i)}, \tag{13}$$

where $v_i(t) = v(t,x_i)$ and $\Delta x_i = x_{i+1} - x_i$. We apply these approximations to Eq. (5). This leads to a set of semi-discrete equations having a matrix form

$$\boldsymbol{v}_t + \boldsymbol{A}v = \boldsymbol{0}, \quad \boldsymbol{A} = \boldsymbol{D} - \boldsymbol{R} \tag{14}$$

where $\boldsymbol{D}$ is a tridiagonal matrix and $\boldsymbol{R}$ is a full matrix resulting from the integral term. The matrix $\boldsymbol{D}$ has the following structure

$$\boldsymbol{D} = \begin{pmatrix} r & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ \boldsymbol{D}_{2,1} & \boldsymbol{D}_{2,2} & \boldsymbol{D}_{2,3} & 0 & \dots & 0 & 0 & 0 \\ 0 & \boldsymbol{D}_{3,2} & \boldsymbol{D}_{3,3} & \boldsymbol{D}_{3,4} & 0 & \dots & 0 & 0 \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 & \dots & 0 \\ 0 & \dots & 0 & \boldsymbol{D}_{i,i-1} & \boldsymbol{D}_{i,i} & \boldsymbol{D}_{i,i+1} & 0 & \vdots \\ 0 & 0 & \dots & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \dots & 0 & \boldsymbol{D}_{n-1,n-2} & \boldsymbol{D}_{n-1,n-1} & \boldsymbol{D}_{n-1,n} \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix}, \tag{15}$$

where the first and last row of the matrix are enforcing the boundary conditions. We use a modified volatility $\hat{\sigma}$ instead of σ defined by

$$\hat{\sigma}^2 = \max\left\{\sigma^2, (r - \lambda\kappa)\frac{\Delta x_i}{x_i}, -(r - \lambda\kappa)\frac{\Delta x_{i-1}}{x_i}\right\}. \tag{16}$$

This artificial volatility ensures that all off-diagonal elements are nonpositive [24]. For the rows $i = 2, \dots, n-1$, the off-diagonal elements of $\boldsymbol{D}$ are given by

$$\boldsymbol{D}_{i,i-1} = \frac{-\hat{\sigma}^2 x_i^2 + (r - \lambda\kappa)x_i\Delta x_i}{\Delta x_{i-1}(\Delta x_{i-1} + \Delta x_i)}, \tag{17}$$

and

$$\boldsymbol{D}_{i,i+1} = \frac{-\hat{\sigma}^2 x_i^2 - (r - \lambda\kappa) x_i \Delta x_{i-1}}{\Delta x_i (\Delta x_{i-1} + \Delta x_i)}. \tag{18}$$

The diagonal elements are given by

$$\boldsymbol{D}_{i,i} = r + \lambda - \boldsymbol{D}_{i,i-1} - \boldsymbol{D}_{i,i+1}, \quad i = 2, \ldots, n-1. \tag{19}$$

3.2. Approximation of the integral term

The integral term

$$I = \int_0^\infty v(t, xy) f(y)\, dy \tag{20}$$

of (5) can be discretized by using the linear interpolation for v between grid points. First, performing a change of variable $y = z/x$, we get

$$I = \int_0^\infty v(t, z) f(z/x)/x\, dz. \tag{21}$$

Now, by using linear interpolation we get an approximation

$$I_i \approx A_i = \sum_{j=1}^{n-1} A_{i,j} \tag{22}$$

of I at each grid point x_i, $i = 2, \ldots, n-1$, where

$$A_{i,j} = \int_{x_j}^{x_{j+1}} \left(\frac{x_{j+1} - z}{\Delta x_j} v(t, x_j) + \frac{z - x_j}{\Delta x_j} v(t, x_{j+1}) \right) f(z/x_i)/x_i\, dz. \tag{23}$$

In the case of Merton's model, the log-normal distribution $f(y) = f_{ln}(y)$ is used, and thus we have

$$A_{i,j} = \frac{1}{\delta\sqrt{2\pi}} \int_{x_j}^{x_{j+1}} \left(\frac{x_{j+1} - z}{\Delta x_j} v(t, x_j) + \frac{z - x_j}{\Delta x_j} v(t, x_{j+1}) \right) \exp\left(-\frac{(\log(z/x_i) - \gamma)^2}{2\delta^2} \right) \Big/ z\, dz. \tag{24}$$

By performing the integration, we obtain

$$\begin{aligned} A_{i,j} = \frac{1}{2\Delta x_j} \Bigg[& \left(\operatorname{erf}\left(\frac{\gamma - \log(x_{j+1}/x_i)}{\delta\sqrt{2}} \right) - \operatorname{erf}\left(\frac{\gamma - \log(x_j/x_i)}{\delta\sqrt{2}} \right) \right) \alpha_j \\ & + \left(\operatorname{erf}\left(\frac{\gamma + \delta^2 - \log(x_{j+1}/x_i)}{\delta\sqrt{2}} \right) - \operatorname{erf}\left(\frac{\gamma + \delta^2 - \log(x_j/x_i)}{\delta\sqrt{2}} \right) \right) x_i \beta_j \Bigg], \end{aligned} \tag{25}$$

where $\operatorname{erf}(\cdot)$ is the error function and

$$\alpha_j = v(t, x_{j+1}) x_j - v(t, x_j) x_{j+1} \quad \text{and} \quad \beta_j = \big(v(t, x_j) - v(t, x_{j+1}) \big) e^{\gamma + \delta^2/2}. \tag{26}$$

In the case of Kou's model, we have the log-double-exponential distribution $f(y) = f_{ld}(y)$. We decompose the integral as $I = I^- + I^+$ and its approximation as $A_i = A_i^- + A_i^+$, $i = 2, \ldots, n-1$, where

$$I^- = \int_0^x v(t, z) f(z/x)_{ld}/x\, dz = q\alpha_2 x^{-\alpha_2} \int_0^x v(t, z) z^{\alpha_2 - 1}\, dz \tag{27}$$

and

$$I^+ = \int_x^\infty v(t, z) f(z/x)_{ld}/x\, dz = p\alpha_1 x^{\alpha_1} \int_x^\infty v(t, z) z^{-\alpha_1 - 1}\, dz. \tag{28}$$

The approximations A_i^- and A_i^+ are given by

$$I_i^- \approx A_i^- = \sum_{j=1}^{i-1} A_{i,j}^- \quad \text{and} \quad I_i^+ \approx A_i^+ = \sum_{j=i}^{n-1} A_{i,j}^+, \tag{29}$$

where

$$A_{i,j}^- = \frac{q x_i^{-\alpha_2}}{(\alpha_2+1)\Delta x_j}\left[\left(x_{j+1}^{\alpha_2+1} - (x_{j+1}+\alpha_2\Delta x_j)x_j^{\alpha_2}\right)v(t,x_j) + \left(x_j^{\alpha 2+1} - (x_j-\alpha_2\Delta x_j)x_{j+1}^{\alpha_2}\right)v(t,x_{j+1})\right] \tag{30}$$

for $j = 1,\dots,i-1$, and

$$A_{i,j}^+ = \frac{p x_i^{\alpha_1}}{(\alpha_1-1)\Delta x_j}\left[\left(x_{j+1}^{1-\alpha_1} - (x_{j+1}+\alpha_1\Delta x_j)x_j^{-\alpha_1}\right)v(t,x_j) + \left(x_j^{1-\alpha 1} - (x_j+\alpha_1\Delta x_j)x_{j+1}^{-\alpha_1}\right)v(t,x_{j+1})\right] \tag{31}$$

for $j = i,\dots,n-1$ are given by performing the integration in (23) with $f(y) = f_{ld}(y)$.

The matrix $\boldsymbol{R}$ resulting from approximating the integral has the sparsity pattern

$$\boldsymbol{R} = \lambda \begin{pmatrix} 0 & \dots & 0 \\ * & \dots & * \\ \vdots & \ddots & \vdots \\ * & \dots & * \\ 0 & \dots & 0 \end{pmatrix}. \tag{32}$$

In the case of Kou's model, the actual computation of $A_{i,j}^-$s and $A_{i,j}^+$s is performed by using recursion formulas given by Toivanen in [24]. With this approach, only $O(n)$ operations are required to calculate the integral approximation. However, this approach is only applicable when a log-double-exponential distribution is used.

3.3. Time discretization

Now for European options we have a semi-discrete linear problem

$$\boldsymbol{v}_t + \boldsymbol{A}\boldsymbol{v} = \boldsymbol{0}. \tag{33}$$

For American options we obtain a semi-discrete LCP

$$\begin{cases} (\boldsymbol{v}_t - \boldsymbol{A}\boldsymbol{v}) \geqslant \boldsymbol{0}, \quad \boldsymbol{v} \geqslant \boldsymbol{g}, \\ (\boldsymbol{v}_t - \boldsymbol{A}\boldsymbol{v})(\boldsymbol{v} - \boldsymbol{g}) = 0. \end{cases} \tag{34}$$

We use the Rannacher scheme [21] for time-discretization; see [12] for convergence analysis. This scheme performs a few first time steps using the implicit Euler scheme, and after that the remaining steps are performed with the Crank–Nicolson method. The implicit Euler steps are taken to avoid possible oscillations due to the nonsmooth final value (payoff function). We perform the first four steps with the implicit Euler method. The time domain $[0,T]$ is split equally with time step $\Delta t = T/m$, with the exception of the implicit Euler steps that are set to have the length $\Delta t/2$.

Applying Rannacher time-stepping scheme to the semi-discrete problem (33) gives us

$$\boldsymbol{B}^{(k)}\boldsymbol{v}^{(k)} = \boldsymbol{b}^{(k)}, \quad k = m+2,\dots,1, \tag{35}$$

where

$$\boldsymbol{B}^{(k)} = \boldsymbol{I} + \theta_k \Delta t_k \boldsymbol{A}, \qquad \boldsymbol{b}^{(k)} = \left(\boldsymbol{I} - (1-\theta_k)\Delta t_k \boldsymbol{A}\right)\boldsymbol{v}^{(k+1)}, \quad \text{and} \quad \boldsymbol{v}^{(m+3)} = \boldsymbol{g}. \tag{36}$$

The parameter θ_k is defined by

$$\theta_k = \begin{cases} 1, & m-1 \leqslant k \leqslant m+2, \\ \frac{1}{2}, & k = m-2,\dots,1. \end{cases} \tag{37}$$

For the semi-discrete LCP (34) the time-stepping gives

$$\begin{cases} \left(\boldsymbol{B}^{(k)}\boldsymbol{v}^{(k)} - \boldsymbol{b}^{(k)}\right) \geqslant \boldsymbol{0}, \quad \boldsymbol{v}^{(k)} \geqslant \boldsymbol{g}, \\ \left(\boldsymbol{B}^{(k)}\boldsymbol{v}^{(k)} - \boldsymbol{b}^{(k)}\right)^T \left(\boldsymbol{v}^{(k)} - \boldsymbol{g}\right) = 0, \end{cases} \quad k = m+2,\dots,1. \tag{38}$$

4. Solution methods

4.1. Iterative method

First we present an iterative method for a generic LCP and a result for its convergence. After that we apply it to price American options. We denote an LCP

$$\boldsymbol{B}\boldsymbol{x} \geqslant \boldsymbol{b}, \qquad \boldsymbol{x} \geqslant \boldsymbol{g}, \qquad (\boldsymbol{B}\boldsymbol{x} - \boldsymbol{b})^T (\boldsymbol{x} - \boldsymbol{g}) = 0 \tag{39}$$

by LCP($\boldsymbol{B}, \boldsymbol{x}, \boldsymbol{b}, \boldsymbol{g}$).

Theorem 1. *Let $\boldsymbol{B}$ be a strictly diagonally dominant square matrix with positive diagonal entries. Let a matrix splitting*

$$\boldsymbol{B} = \boldsymbol{T} - \boldsymbol{J} \tag{40}$$

be such that the inequality

$$\boldsymbol{T}_{i,i} - \sum_{j \neq i} |\boldsymbol{T}_{i,j}| - \sum_{j} |\boldsymbol{J}_{i,j}| > 0 \tag{41}$$

holds for all i. Then the vectors $\boldsymbol{x}^{l+1}$, $l = 0, 1, 2, \ldots$, defined by the iteration

$$\mathrm{LCP}\big(\boldsymbol{T}, \boldsymbol{x}^{l+1}, \boldsymbol{J}\boldsymbol{x}^{l} + \boldsymbol{b}, \boldsymbol{g}\big), \tag{42}$$

and an initial guess $\boldsymbol{x}^0$ converge to the solution $\boldsymbol{x}$ of the LCP (39)*. Furthermore, for the error $\boldsymbol{e}^l = \boldsymbol{x}^l - \boldsymbol{x}$ the norm inequality*

$$\big\|\boldsymbol{e}^{l+1}\big\|_\infty \leqslant \left(\max_i \frac{\sum_j |\boldsymbol{J}_{i,j}|}{\boldsymbol{T}_{i,i} - \sum_{j \neq i} |\boldsymbol{T}_{i,j}|} \right) \big\|\boldsymbol{e}^{l}\big\|_\infty \tag{43}$$

holds.

Proof. Substituting $\boldsymbol{x}^{l+1} = \boldsymbol{x} + \boldsymbol{e}^{l+1}$ and $\boldsymbol{x}^l = \boldsymbol{x} + \boldsymbol{e}^l$ to (42) gives us for the error $\boldsymbol{e}^{l+1}$ an LCP

$$\begin{aligned} &\boldsymbol{T}\boldsymbol{e}^{l+1} \geqslant \boldsymbol{J}\boldsymbol{e}^{l} - \boldsymbol{B}\boldsymbol{x} + \boldsymbol{b}, \qquad \boldsymbol{e}^{l+1} \geqslant \boldsymbol{g} - \boldsymbol{x}, \\ &\big(\boldsymbol{T}\boldsymbol{e}^{l+1} - \boldsymbol{J}\boldsymbol{e}^{l} + \boldsymbol{B}\boldsymbol{x} - \boldsymbol{b}\big)^T \big(\boldsymbol{e}^{l+1} - \boldsymbol{g} + \boldsymbol{x}\big) = 0. \end{aligned} \tag{44}$$

Let the ith component of $\boldsymbol{e}^{l+1}$ denoted by $\boldsymbol{e}_i^{l+1}$ have the largest absolute value, that is, $\|\boldsymbol{e}^{l+1}\|_\infty = |\boldsymbol{e}_i^{l+1}| \geqslant |\boldsymbol{e}_j^{l+1}|$ for all j. In the following, we need an estimate of $|(\boldsymbol{J}\boldsymbol{e}^l)_i|$ from above given by

$$\big|\big(\boldsymbol{J}\boldsymbol{e}^{l}\big)_i\big| = \left|\sum_j \boldsymbol{J}_{i,j}\boldsymbol{e}_j^l\right| \leqslant \sum_j |\boldsymbol{J}_{i,j}||\boldsymbol{e}_j^l| \leqslant \left(\sum_j |\boldsymbol{J}_{i,j}|\right) \big\|\boldsymbol{e}^{l}\big\|_\infty. \tag{45}$$

Let us consider the ith inequalities and equations of the LCPs (39) and (44). We have the following four possibilities:

1. $(\boldsymbol{B}\boldsymbol{x} - \boldsymbol{b})_i = 0$ and $(\boldsymbol{T}\boldsymbol{e}^{l+1} - \boldsymbol{J}\boldsymbol{e}^{l} + \boldsymbol{B}\boldsymbol{x} - \boldsymbol{b})_i = 0$: We have $(\boldsymbol{T}\boldsymbol{e}^{l+1})_i = (\boldsymbol{J}\boldsymbol{e}^{l})_i$ and also

$$\big|\big(\boldsymbol{T}\boldsymbol{e}^{l+1}\big)_i\big| = \big|\big(\boldsymbol{J}\boldsymbol{e}^{l}\big)_i\big|. \tag{46}$$

We estimate $|(\boldsymbol{T}\boldsymbol{e}^{l+1})_i|$ from below as follows:

$$\begin{aligned} \big|\big(\boldsymbol{T}\boldsymbol{e}^{l+1}\big)_i\big| &= \left|\boldsymbol{T}_{i,i}\boldsymbol{e}_i^{l+1} + \sum_{j \neq i} \boldsymbol{T}_{i,j}\boldsymbol{e}_j^{l+1}\right| \geqslant |\boldsymbol{T}_{i,i}\boldsymbol{e}_i^{l+1}| - \left|\sum_{j \neq i} \boldsymbol{T}_{i,j}\boldsymbol{e}_j^{l+1}\right| \\ &\geqslant \boldsymbol{T}_{i,i}\big|\boldsymbol{e}_i^{l+1}\big| - \sum_{j \neq i} |\boldsymbol{T}_{i,j}|\big|\boldsymbol{e}_j^{l+1}\big| \geqslant \left(\boldsymbol{T}_{i,i} - \sum_{j \neq i} |\boldsymbol{T}_{i,j}|\right)\big|\boldsymbol{e}_i^{l+1}\big|. \end{aligned} \tag{47}$$

By combining the estimates (45) and (47) with Eq. (46), we obtain

$$\big\|\boldsymbol{e}^{l+1}\big\|_\infty \leqslant \frac{\sum_j |\boldsymbol{J}_{i,j}|}{\boldsymbol{T}_{i,i} - \sum_{j \neq i} |\boldsymbol{T}_{i,j}|} \big\|\boldsymbol{e}^{l}\big\|_\infty. \tag{48}$$

2. $(\boldsymbol{Bx}-\boldsymbol{b})_i = 0$ and $\boldsymbol{e}_i^{l+1} - \boldsymbol{g}_i + \boldsymbol{x}_i = 0$: We have

$$-\big(\boldsymbol{T}\boldsymbol{e}^{l+1}\big)_i \leqslant -\big(\boldsymbol{J}\boldsymbol{e}^{l} - \boldsymbol{B}\boldsymbol{x} + \boldsymbol{b}\big)_i = -\big(\boldsymbol{J}\boldsymbol{e}^{l}\big)_i \leqslant \big|\big(\boldsymbol{J}\boldsymbol{e}^{l}\big)_i\big|. \tag{49}$$

We note that $\boldsymbol{e}_i^{l+1} = \boldsymbol{g}_i - \boldsymbol{x}_i \leqslant 0$. We estimate $-(\boldsymbol{T}\boldsymbol{e}^{l+1})_i$ from below as follows:

$$\begin{aligned} -\big(\boldsymbol{T}\boldsymbol{e}^{l+1}\big)_i &= -\boldsymbol{T}_{i,i}\boldsymbol{e}_i^{l+1} - \sum_{j\neq i}\boldsymbol{T}_{i,j}\boldsymbol{e}_j^{l+1} \geqslant \boldsymbol{T}_{i,i}\boldsymbol{e}_i^{l+1} - \left|\sum_{j\neq i}\boldsymbol{T}_{i,j}\boldsymbol{e}_j^{l+1}\right| \\ &\geqslant \boldsymbol{T}_{i,i}\big|\boldsymbol{e}_i^{l+1}\big| - \sum_{j\neq i}|\boldsymbol{T}_{i,j}|\big|\boldsymbol{e}_j^{l+1}\big| \geqslant \left(\boldsymbol{T}_{i,i} - \sum_{j\neq i}|\boldsymbol{T}_{i,j}|\right)\big|\boldsymbol{e}_i^{l+1}\big|. \end{aligned} \tag{50}$$

By combining the estimates (45) and (50) with the inequality (49), we obtain again the inequality (48).

3. $\boldsymbol{x}_i - \boldsymbol{g}_i = 0$ and $(\boldsymbol{T}\boldsymbol{e}^{l+1} - \boldsymbol{J}\boldsymbol{e}^{l} + \boldsymbol{B}\boldsymbol{x} - \boldsymbol{b})_i = 0$: Using the inequality $(\boldsymbol{B}\boldsymbol{x}-\boldsymbol{b})_i \geqslant 0$, we obtain

$$\big(\boldsymbol{T}\boldsymbol{e}^{l+1}\big)_i = \big(\boldsymbol{J}\boldsymbol{e}^{l} - \boldsymbol{B}\boldsymbol{x} + \boldsymbol{b}\big)_i = \big(\boldsymbol{J}\boldsymbol{e}^{l}\big)_i - (\boldsymbol{B}\boldsymbol{x}-\boldsymbol{b})_i \leqslant \big(\boldsymbol{J}\boldsymbol{e}^{l}\big)_i \leqslant \big|\big(\boldsymbol{J}\boldsymbol{e}^{l}\big)_i\big|. \tag{51}$$

We note that $\boldsymbol{e}_i^{l+1} = \boldsymbol{e}_i^{l+1} - \boldsymbol{g}_i + \boldsymbol{x}_i \geqslant 0$. We estimate $(\boldsymbol{T}\boldsymbol{e}^{l+1})_i$ from below as follows:

$$\begin{aligned} \big(\boldsymbol{T}\boldsymbol{e}^{l+1}\big)_i &= \boldsymbol{T}_{i,i}\boldsymbol{e}_i^{l+1} + \sum_{j\neq i}\boldsymbol{T}_{i,j}\boldsymbol{e}_j^{l+1} \geqslant \boldsymbol{T}_{i,i}\boldsymbol{e}_i^{l+1} - \left|\sum_{j\neq i}\boldsymbol{T}_{i,j}\boldsymbol{e}_j^{l+1}\right| \\ &\geqslant \boldsymbol{T}_{i,i}\big|\boldsymbol{e}_i^{l+1}\big| - \sum_{j\neq i}|\boldsymbol{T}_{i,j}|\big|\boldsymbol{e}_j^{l+1}\big| \geqslant \left(\boldsymbol{T}_{i,i} - \sum_{j\neq i}|\boldsymbol{T}_{i,j}|\right)\big|\boldsymbol{e}_i^{l+1}\big|. \end{aligned} \tag{52}$$

By combining the estimates (45) and (52) with the inequality (51), we obtain again the inequality (48).

4. $\boldsymbol{x}_i - \boldsymbol{g}_i = 0$ and $\boldsymbol{e}_i^{l+1} - \boldsymbol{g}_i + \boldsymbol{x}_i = 0$: We have $\|\boldsymbol{e}^{l+1}\|_\infty = |\boldsymbol{e}_i^{l+1}| = 0$.

Thus, in all four possible cases the norm inequality (48) holds for the error $\boldsymbol{e}^{l+1}$. Taking maximum of the factor in the right-hand side of (48) over i gives us theorem's norm inequality (43).

From the assumption (41) for the matrix splitting (40) it follows that the factor in the right-hand side of (43) is strictly less than one. Thus, the error $\boldsymbol{e}^{l+1}$ converges to zero and $\boldsymbol{x}^{l+1}$ converges to $\boldsymbol{x}$. □

A stationary iterative method was proposed by Tavella and Randall in [23] for pricing European options under jump-diffusion models. This iteration was analyzed by d'Halluin, Forsyth, and Vetzal in [9]. We adopt the formulation presented by Almendral and Oosterlee in [2], where the matrix $\boldsymbol{B}^{(k)}$ is split with a regular splitting

$$\boldsymbol{B}^{(k)} = \boldsymbol{T} - \boldsymbol{J}, \quad \text{where } \boldsymbol{T} = \boldsymbol{I} + \theta_k \Delta t_k \boldsymbol{D} \text{ and } \boldsymbol{J} = -\theta_k \Delta t_k \boldsymbol{R}. \tag{53}$$

For European options, the iterative method reads

$$\boldsymbol{v}^{l+1} = \boldsymbol{T}^{-1}\big(\boldsymbol{b}^{(k)} + \boldsymbol{J}\boldsymbol{v}^{l}\big), \quad l = 0, 1, \ldots, \tag{54}$$

where the initial guess $\boldsymbol{v}^0$ is taken to be $\boldsymbol{v}^{(k+1)}$. Each iteration requires a solution with the tridiagonal $\boldsymbol{T}$, and the multiplication of a vector by $\boldsymbol{J}$.

We can use the same matrix splitting (53) with the iteration (42) for American options. Let us consider the convergence of this iteration and more specifically the coefficient in the inequality (43). As the first and last row of the matrix $\boldsymbol{R}$ contains only zeroes the value of

$$\frac{\sum_j |\boldsymbol{J}_{i,j}|}{\boldsymbol{T}_{i,i} - \sum_{j\neq i}|\boldsymbol{T}_{i,j}|} \tag{55}$$

is zero for $i = 1$ and $i = n$. In the following, we consider i in the range $2, \ldots, n-1$. We have

$$\boldsymbol{T}_{i,i} - \sum_{j\neq i}|\boldsymbol{T}_{i,j}| = 1 + \theta_k \Delta t_k (\boldsymbol{D}_{i,i} + \boldsymbol{D}_{i,i-1} + \boldsymbol{D}_{i,i+1}) = 1 + \theta_k \Delta t_k (r + \lambda). \tag{56}$$

For our quadrature rules, it holds

$$\sum_j \boldsymbol{R}_{i,j} = \lambda \int_0^{x_n} f(z/x_i)/x_i \, dz = \lambda \int_0^{x_n/x_i} f(y)\, dy < \lambda. \tag{57}$$

Table 1
Reference prices used for numerical experiments.

Model and type	Value at 90	Value at 100	Value at 110
Kou, European put [24]	9.430457	2.731259	0.552363
Kou, American put [24]	10.005071	2.807879	0.561876
Merton, European call [9]	0.527638	4.391246	12.643406
Merton, American put	10.003815	3.241215	1.419796

Thus, we have

$$\sum_j |\boldsymbol{J}_{i,j}| = \theta_k \Delta t_k \sum_j \boldsymbol{R}_{i,j} < \theta_k \Delta t_k \lambda. \tag{58}$$

The above observations, gives us the following result based on Theorem 1.

Corollary 1. *With the above described discretizations and splitting* (53) *of* $\boldsymbol{B}^{(k)}$, *for the error* $\boldsymbol{e}^l$ *in the iteration* (42) *the inequality*

$$\left\|\boldsymbol{e}^{l+1}\right\|_\infty < C\left\|\boldsymbol{e}^l\right\|_\infty \tag{59}$$

holds, where the constant is

$$C = \frac{\theta_k \Delta t_k \lambda}{1 + \theta_k \Delta t_k (r + \lambda)}. \tag{60}$$

Furthermore, for any $r \geqslant 0$ *and* $\lambda \geqslant 0$, $C < 1$ *holds, that is, the iteration converges.*

4.2. Brennan and Schwartz algorithm

To solve the resulting LCP (42) with a tridiagonal $\boldsymbol{T}$ in each iteration, we use the Brennan and Schwartz algorithm. The original algorithm presented by Brennan and Schwartz in [6], is based on Gaussian elimination; see also [1,15]. Consider a linear complementarity problem LCP($\boldsymbol{T}, \boldsymbol{v}, \boldsymbol{b}, \boldsymbol{g}$). We form an $\boldsymbol{U}\boldsymbol{L}$-decomposition of $\boldsymbol{T}$ such that

$$\boldsymbol{U}\boldsymbol{L} = \boldsymbol{T}, \tag{61}$$

and select the diagonal of $\boldsymbol{L}$ to consist of ones. The algorithm now reads

$$\begin{aligned}
&\boldsymbol{y} = \boldsymbol{U}^{-1}\boldsymbol{b} \\
&v_1 = \max\{y_1, g_1\} \\
&\text{DO} \quad i = 2, \ldots, n \\
&\quad v_i = \max\{y_i - \boldsymbol{L}_{i,i-1} v_{i-1}, g_i\} \\
&\text{END DO}
\end{aligned} \tag{62}$$

where $\boldsymbol{L}_{i,j}$ are components of $\boldsymbol{L}$. Note that an American call option can be priced with the same approach by reversing the order of the Brennan and Schwartz algorithm, assuming the underlying asset pays dividends continuously; see [15] for the algorithm in reverse order.

5. Numerical experiments

5.1. European put option under Kou's model

First, we price European put options under Kou's model using the following model parameters:

$$\begin{aligned}
&\sigma = 0.15, \quad && r = 0.05, \quad && T = 0.25, \quad && K = 100, \\
&\lambda = 0.1, \quad && \alpha_1 = 3.0465, \quad && \alpha_2 = 3.0775, \quad && p = 0.3445.
\end{aligned} \tag{63}$$

These parameters are also used by d'Halluin, Forsyth and Vetzal in [9] and Toivanen in [24]. We use the reference prices described in [24]. Table 1 gives a complete list of reference prices used in this section. A uniform space grid between $[0, X]$ is used with n nodes and $X = 400$. We continue the iteration until $\|\boldsymbol{v}^l - \boldsymbol{v}^{l+1}\|_2$ is less than 10^{-8}. This stopping criterion is used in all the examples.

The results of the pricing are given in Table 2, which reports pricing errors, the number of total iterations and execution times. Also the ratio of the consecutive errors in the l_2-norm at $x = 90$, $x = 100$, and $x = 110$. The computations were performed on a PC with a 2.2 GHz AMD Athlon 3500+ processor. The codes were run on Matlab, however time consuming parts were implemented as mex files using C language. Second-order convergence is evident from Table 2 as the ratio is close to 4.

Table 2
European put option pricing errors, execution times, total iterations and the ratios of errors under Kou's model.

n	m	Error at 90	Error at 100	Error at 110	Ratio	Iterations	Time (ms)
50	20	3.061e−1	4.459e−1	1.766e−1		66	0.9
100	40	4.512e−2	−1.116e−1	1.047e−2	4.71	126	1.3
200	80	−2.441e−3	−2.648e−2	−5.386e−3	4.45	246	3.6
400	160	−6.598e−4	−6.550e−3	−1.377e−3	4.03	324	9.6
800	320	−1.678e−4	−1.634e−3	−3.462e−4	4.01	644	81.4
1600	640	−4.199e−5	−4.084e−4	−8.685e−5	4.00	1284	157.9

Table 3
American put option pricing errors, execution times, total iterations and the ratios of errors under Kou's model.

n	m	Error at 90	Error at 100	Error at 110	Ratio	Iterations	Time (ms)
50	20	2.436e−1	4.279e−1	1.757e−1		66	0.8
100	40	2.987e−2	−1.201e−1	1.064e−2	4.21	126	1.3
200	80	−5.071e−3	−2.989e−2	−5.767e−3	4.03	193	3.2
400	160	−4.263e−3	−7.623e−3	−1.547e−3	3.48	324	10.1
800	320	−3.123e−4	−1.964e−3	−4.126e−4	4.37	644	88.0
1600	640	−1.003e−4	−5.090e−4	−1.106e−4	3.93	1284	168.4

Table 4
European call option pricing errors, execution times, total iterations and the ratios of errors under Merton's model with a straightforward matrix–vector multiplication.

n	m	Error at 90	Error at 100	Error at 110	Ratio	Iterations	Time (ms)
50	20	3.845e−1	4.025e−1	1.060e−1		66	10.0
100	40	6.102e−2	−1.172e−1	7.445e−4	4.29	126	77.1
200	80	−1.801e−3	−2.778e−2	−5.746e−3	4.65	246	221.2
400	160	−5.144e−4	−6.873e−3	−1.464e−3	4.03	482	1197.6
800	320	−1.325e−4	−1.714e−3	−3.677e−4	4.01	644	5996.0
1600	640	−3.336e−5	−4.285e−4	−9.215e−5	4.00	1284	37 893.7

5.2. American put option under Kou's model

In this second example, we price American put options with the same parameters as in the previous example. This problem has been considered in [24,25] and the reference prices are taken from [24]. The Brennan and Schwartz algorithm is used to solve the resulting LCPs.

Results are listed in Table 3. Errors, execution times and iteration counts are similar to the previous example with European options. The iteration converges rapidly with typically two iterations on each time step. Rapid convergence can be also expected based on Corollary 1 as the value of C in (59) is about 6.24e−4 for $m = 20$ and 1.95e−5 for $m = 640$. Accuracy is similar to the European option, which could be further increased with a nonuniform grid. Note that we described the discretization (12) and (13) also for nonuniform grids. Execution times are only slightly higher when compared to the previous example. A second-order convergence is obtained.

5.3. European call option under Merton's model

In the third example, we price European call options under Merton's model. We use the following parameters:

$$\begin{aligned} &\sigma = 0.15, \qquad r = 0.05, \qquad T = 0.25, \qquad K = 100, \\ &\lambda = 0.1, \qquad \gamma = -0.9, \qquad \delta = 0.45. \end{aligned} \tag{64}$$

These parameters are also used by d'Halluin, Forsyth, and Vetzal in [9]. They are equal to the ones used in the previous examples with the exception of the distribution parameters γ and δ. The same reference prices were used as in [9].

Instead of pricing a call option directly, we compute the price of a put option with the same strike price, and then use the put–call parity

$$v_c(t,x) = v_p(t,x) + x - Ke^{-r(T-t)} \tag{65}$$

to obtain the price of the call option. Above v_c is the price of the call option and v_p is the price of the put option.

Results are listed in Table 4. Again, second-order convergence is observed, and similar errors and iteration counts as in previous examples. However, this time execution times are significantly higher. This is to be expected, and is due to vector–matrix multiplication with a full $n \times n$ matrix that is required to compute the integral approximation. A much faster implementation can be done with FFT. With this approach the number of required operations to compute the integral

Table 5
European call option pricing errors, execution times, total iterations and the ratios of errors under Merton's model and the FFT approach.

n	m	Error at 90	Error at 100	Error at 110	Ratio	Iterations	Time (ms)
50	20	3.752e−1	3.949e−1	9.971e−2		66	2.0
100	40	5.919e−2	−1.186e−1	−5.002e−4	4.18	126	6.2
200	80	−2.142e−3	−2.809e−2	−6.018e−3	4.60	246	48.8
400	160	−5.945e−4	−6.943e−3	−1.527e−3	4.04	482	121.1
800	320	−1.528e−4	−1.732e−3	−3.840e−4	4.01	644	301.2
1600	640	−3.864e−5	−4.332e−4	−9.643e−5	4.00	1284	1215.7

Table 6
American put option pricing errors, execution times, total iterations and the ratios of errors under Merton's model with straightforward matrix–vector multiplication.

n	m	Error at 90	Error at 100	Error at 110	Ratio	Iterations	Time (ms)
50	20	2.846e−1	3.722e−1	1.043e−1		66	9.9
100	40	3.492e−2	−1.286e−1	7.778e−4	3.60	126	84.9
200	80	−3.815e−3	−3.210e−2	−6.143e−3	4.05	246	220.8
400	160	−3.815e−3	−8.166e−3	−1.625e−3	3.59	324	1007.7
800	320	−8.542e−4	−2.067e−3	−4.204e−4	4.03	644	5975.4
1600	640	−2.840e−4	−5.063e−4	−1.047e−4	3.86	1284	38 313.7

Table 7
American put option pricing errors, execution times, total iterations and the ratios of errors under Merton's model and the FFT approach.

n	m	Error at 90	Error at 100	Error at 110	Ratio	Iterations	Time (ms)
50	20	2.837e−1	3.684e−1	1.007e−1		66	2.0
100	40	3.468e−2	−1.295e−1	−1.564e−4	3.55	126	6.0
200	80	−3.815e−3	−3.235e−2	−6.388e−3	4.04	246	43.9
400	160	−3.815e−3	−8.226e−3	−1.685e−3	3.60	324	103.7
800	320	−8.559e−4	−2.083e−3	−4.361e−4	4.02	644	305.8
1600	640	−2.843e−4	−5.104e−4	−1.089e−4	3.86	1284	1221.0

approximation can be reduced from $O(n^2)$ to $O(n \log n)$; see [2,9,22] for example. This is almost as good as the $O(n)$ operations required by the recursion formulas used in the first two examples.

Results computed with the FFT approach are listed in Table 5 for comparison purposes. To avoid wrap-around errors, we have used the embedding of a Toeplitz matrix into a circulant matrix described by Almendral and Oosterlee in [2]. Standard FFT methods require a uniform grid, which in this case is in the $\log x$ space. An interpolation step is required to move the data to a uniform grid in the $\log x$ space, and then another interpolation step is needed to move the result back to the grid in the x space. These steps explain the slight deterioration of accuracy that is evident. However, this is a small price to pay for such a significant performance increase. See [9] for more details on the interpolation.

5.4. American put option under Merton's model

In this example, we price American put options under Merton's model. The same parameters are used as in the previous example. The reference prices were computed numerically using a fine grid of $n = 6400$ and $m = 2560$. Results are reported in Table 6. The errors, the ratios of errors, iteration counts and execution times are similar to the European counterpart. Again, the results computed with the FFT approach are listed in Table 7 for comparison purposes.

6. Conclusions

We described an efficient iteration to solve LCPs resulting from the implicit finite difference discretization of PIDEs for pricing American options under jump-diffusion models. It is a generalization of the stationary iterative method for European options presented by Tavella and Randall in [23]. The iteration requires solving LCPs with a banded matrix instead of the full matrix resulting from the discretization of the jump model. We proved that the iteration converges and gave the convergence rate in l_∞-norm.

We considered Kou's model [17] and Merton's model [20] for pricing American and European options. The resulting LCPs in the iteration for American options are tridiagonal and they were solved with the Brennan and Schwartz algorithm [6]. European options were priced with the iteration described by Tavella and Randall in [23]. For both American and European options, the iterations converged rapidly with typically two iterations per time step.

A reasonably accurate solution can be computed in a few milliseconds, assuming the integral approximation can be efficiently computed. For Kou's model, recursion formulas introduced by Toivanen in [24] were used, and for Merton's model an FFT based approach was used to achieve good performance. Under both models, second-order convergence was observed.

Acknowledgements

The research was supported by the Academy of Finland, grants 121271 and 129690.

References

[1] Y. Achdou, O. Pironneau, Computational Methods for Option Pricing, Frontiers in Applied Mathematics, vol. 30, SIAM, Philadelphia, PA, 2005.
[2] A. Almendral, C.W. Oosterlee, Numerical valuation of options with jumps in the underlying, Appl. Numer. Math. 53 (2005) 1–18.
[3] L. Andersen, J. Andreasen, Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing, Rev. Derivatives Res. 4 (2000) 231–262.
[4] D.S. Bates, Jumps and stochastic volatility: Exchange rate processes implicit Deutsche mark options, Rev. Finan. Stud. 9 (1996) 69–107.
[5] F. Black, M. Scholes, The pricing of options and corporate liabilities, J. Polit. Economy 81 (1973) 637–654.
[6] M.J. Brennan, E.S. Schwartz, The valuation of American put options, J. Finance 32 (1977) 449–462.
[7] P. Carr, H. Geman, D.B. Madan, M. Yor, The fine structure of asset returns: an empirical investigation, J. Business 75 (2002) 305–332.
[8] Y. d'Halluin, P.A. Forsyth, G. Labahn, A penalty method for American options with jump diffusion processes, Numer. Math. 97 (2004) 321–352.
[9] Y. d'Halluin, P.A. Forsyth, K.R. Vetzal, Robust numerical methods for contingent claims under jump diffusion processes, IMA J. Numer. Anal. 25 (2005) 87–112.
[10] B. Eraker, M. Johannes, N. Polson, The impact of jumps in volatility and returns, J. Finance 58 (2003) 1269–1300.
[11] F. Fang, C.W. Oosterlee, Pricing early-exercise and discrete barrier options by Fourier-cosine series expansions, Numer. Math. 114 (2009) 27–62.
[12] M.B. Giles, R. Carter, Convergence analysis of Crank–Nicolson and Rannacher time-marching, J. Comput. Finance 9 (2006) 89–112.
[13] J. Huang, J.S. Pang, Option pricing and linear complementarity, J. Comput. Finance 2 (1998) 31–60.
[14] S. Ikonen, J. Toivanen, Operator splitting methods for American option pricing, Appl. Math. Lett. 17 (2004) 809–814.
[15] S. Ikonen, J. Toivanen, Pricing American options using LU decomposition, Appl. Math. Sci. 1 (2007) 2529–2551.
[16] S. Ikonen, J. Toivanen, Operator splitting methods for pricing American options under stochastic volatility, Numer. Math. 113 (2009) 299–324.
[17] S.G. Kou, A jump-diffusion model for option pricing, Manage. Sci. 48 (2002) 1086–1101.
[18] R. Lord, F. Fang, F. Bervoets, C.W. Oosterlee, A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes, SIAM J. Sci. Comput. 30 (2008) 1678–1705.
[19] R.C. Merton, Theory of rational option pricing, Bell J. Econom. Manage. Sci. 4 (1973) 141–183.
[20] R.C. Merton, Option pricing when underlying stock returns are discontinuous, J. Finan. Econ. 3 (1976) 125–144.
[21] R. Rannacher, Finite element solution of diffusion problems with irregular data, Numer. Math. 43 (1984) 309–327.
[22] E.W. Sachs, A.K. Strauss, Efficient solution of a partial integro-differential equation in finance, Appl. Numer. Math. 58 (2008) 1687–1703.
[23] D. Tavella, C. Randall, Pricing Financial Instruments: The Finite Difference Method, John Wiley & Sons, Chichester, 2000.
[24] J. Toivanen, Numerical valuation of European and American options under Kou's jump-diffusion model, SIAM J. Sci. Comput. 30 (2008) 1949–1970.
[25] J. Toivanen, A high-order front-tracking finite difference method for pricing American options under jump-diffusion models, J. Comput. Finance 13 (2010) 61–79.
[26] P. Wilmott, Derivatives, John Wiley & Sons Ltd., Chichester, 1998.

PIII

ROBUST AND EFFICIENT IMEX SCHEMES FOR OPTION PRICING UNDER JUMP-DIFFUSION MODELS

by

S. Salmi and J. Toivanen 2013

Submitted for publication

Robust and efficient IMEX schemes for option pricing under jump-diffusion models

Santtu Salmi* Jari Toivanen*†

October 17, 2013

Abstract

We propose families of IMEX time discretization schemes for the partial integro-differential equation derived for the pricing of options under a jump-diffusion process. The schemes include the families of IMEX-midpoint, IMEX-CNAB and IMEX-BDF2 schemes. Each family is defined by a convex combination parameter $c \in [0, 1]$, which divides the zeroth-order term due to the jumps between the implicit and explicit part in the time discretization. These IMEX schemes lead to tridiagonal systems, which can be solved extremely efficiently. The schemes are studied through Fourier stability analysis and numerical experiments. It is found that, under suitable assumptions and time step restrictions, the IMEX-midpoint family is absolutely stable only for $c = 0$, while the IMEX-CNAB and the IMEX-BDF2 families are absolutely stable for all $c \in [0, 1]$. The IMEX-CNAB $c = 0$ scheme produced the smallest error in our numerical experiments.

1 Introduction

Jump-diffusion models have become an important modeling tool in the pricing of financial derivatives. In his seminal paper [17] in 1976, Merton proposed the addition of jumps into the Black-Scholes model. Contrary to models with continuous paths, jump-diffusion models allow large sudden changes in the price of the underlying asset. The possibility of jumps is particularly important for options with short maturities, since large sudden changes under a purely Brownian motion type process have almost zero probability. Indeed, the implied volatility is known to exhibit a smile like shape with respect to the strike price, which becomes more pronounced near the expiry of the option. Jump-diffusion models have been shown to naturally produce implied volatilities with a smile [1, 5].

*Department of Mathematical Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland. Email: santtu.salmi@jyu.fi. Salmi acknowledges financial support from the Finnish Academy of Science and Letters.

†Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305, USA. Email: toivanen@stanford.edu

Under the Merton model, jumps follow a compound Poisson process with constant intensity and log-normal jump sizes. Other well-known jump-diffusion models in the literature include the Kou model [14] and the more general CGMY model [4]. The CGMY model allows infinite jump activity, while the other mentioned models are finite activity jump-diffusion models. Here we restrict to finite activity jump-diffusion models. However, we note that in numerical computations it is usual practice to reduce the infinite activity jump-diffusion model to a finite activity jump-diffusion case by approximating the infinite activity portion of the jump distribution by artificial diffusion; see [6], for example. Thus, the results presented here apply also in that case.

A partial integro-differential equation (PIDE) can be derived for the price of a European option under a jump-diffusion process. The American option pricing problem can be formulated in the form of an inequality by adopting the linear complementarity problem (LCP) formulation. We will consider the time discretization schemes for such problems. Second-order accurate time discretization schemes are traditionally implicit schemes. However, employing standard implicit time discretization schemes on jump-diffusion models results in systems with full matrices. A more efficient approach is to employ iterative methods, for example, a fixed-point iteration for European options [21], or a similar iteration for American options [19]. An efficient penalty iteration was also proposed for American options in [7]. The iterative methods enable the use of standard implicit discretizations, which are accurate and unconditionally stable.

Alternatively, the inversion of a full matrix can be avoided by employing implicit-explicit (IMEX) discretizations, where typically the jump term is treated explicitly and the rest implicitly. The IMEX discretizations lead to tridiagonal systems. The tridiagonal systems can then be solved directly and extremely efficiently; see [20] for a comparison. These discretization schemes are *conditionally* stable. A first-order accurate IMEX-Euler scheme for jump-diffusion models was formulated in [6]. Second-order accurate IMEX-schemes are typically two-step linear multistep methods, such as the IMEX-midpoint scheme; see [15,16] for example. For European options even more efficient numerical methods exist, which exploit the linear nature of the European option pricing problem, such as the Runge-Kutta type methods proposed in [3,9], or the specialized Fourier transform based methods such as in [8]. For American options, however, second-order accurate schemes are generally the most efficient methods available.

An interesting question is the temporal discretization of the zeroth-order term λv in the PIDE. In [15,16], the authors propose an IMEX-midpoint scheme where they include the zeroth-order term explicitly in the time discretization. Another obvious choice is to include the zeroth-order term in the implicit part of an IMEX scheme. However, as the jump term and the zeroth-order term partially cancel each other, it begs the question if including a portion of the zeroth-order term in the explicit part would improve the performance and stability of the discretization method by essentially canceling a portion of the explicit part. Here we will introduce a convex combination of the zeroth-order term into various IMEX schemes and investigate

the effects through Fourier stability analysis and numerical experiments. The optimal weights for this convex combination to minimize the explicit part (i.e. explicit eigenvalue) would require the exact knowledge of the eigenvalues of the discretized problem. Unfortunately, the eigenvalues are generally not available in closed form, and we will therefore consider some a priori chosen convex combinations.

2 Preliminaries

In this section, we will describe the mathematical model for pricing European options under a jump-diffusion process. We assume that the stock price S_t follows a finite activity jump-diffusion process as in [5]. Then the stochastic differential equation for S_t is given by

$$\frac{dS_t}{S_{t-}} = (\mu - \lambda\kappa)dt + \sigma dW(t) + (\eta - 1)dN_t, \tag{1}$$

where μ is the drift rate, σ is the volatility, $W(t)$ is a standard Brownian motion, N_t is a Poisson process with rate λ, $(\eta-1)$ is a random variable of the jump distribution producing a jump from S_{t-} to ηS_{t-}, and κ is the expected value of $(\eta - 1)$. The notation S_{t-} means that whenever there is a jump, the value of the process before the jump is used.
Now, let $v(x, \tau)$ be the value of a European option contract that depends on the price of the underlying asset $x = S_t$ and the time to maturity $\tau = T - t$. The function $v(x, \tau)$ satisfies the PIDE

$$v_\tau = \frac{1}{2}\sigma^2 x^2 v_{xx} + (r - \lambda\kappa)xv_x - (r + \lambda)v + \lambda\left(\int_0^\infty v(xy, \tau)f(y)dy\right) =: Lv, \tag{2}$$

for all $(x, \tau) \in [0, \infty) \times (0, T]$, where r is the risk-free interest rate, and f is the density function of the jump distribution [17, 23]. The value of v at the maturity is given by

$$v(x, 0) = g(x), \quad x \in [0, \infty), \tag{3}$$

where $g(x)$ is the payoff function for the option contract. For a put option, it is

$$g(x) = \max(K - x, 0), \tag{4}$$

where K is the strike price. The boundary conditions for a European put option are given by

$$\begin{aligned} v(0, \tau) &= Ke^{-r\tau}, \\ \lim_{x\to\infty} v(x, \tau) &= 0, \qquad \tau \in [0, T]. \end{aligned} \tag{5}$$

To simplify notation, we split the PIDE (2) into parts as follows

$$Lv = Dv + \lambda(Jv - v), \tag{6}$$

where D is the differential operator and J is the integral operator defined by

$$\begin{aligned} Dv &:= \frac{1}{2}\sigma^2 x^2 v_{xx} + (r - \lambda\kappa)xv_x - rv, \\ Jv &:= \left(\int_0^\infty v(xy,\tau) f(y) dy \right). \end{aligned} \tag{7}$$

3 Time Stepping Schemes and Stability

In the following we consider the stability properties of IMEX schemes with three time levels in the context of the PIDE (2). The considered schemes include the IMEX-midpoint (or IMEX-CNLF, Crank-Nicolson, Leap-Frog) scheme

$$\frac{\boldsymbol{v}_{m+1} - \boldsymbol{v}_{m-1}}{2} = \Delta\tau\lambda(\boldsymbol{J} - c\boldsymbol{I})\boldsymbol{v}_m + \Delta\tau(\boldsymbol{D} - \lambda(1-c)\boldsymbol{I})\frac{\boldsymbol{v}_{m+1} + \boldsymbol{v}_{m-1}}{2}, \tag{8}$$

where $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{D}$ and $\boldsymbol{J}$ are matrices resulting from the spatial discretization of (7). We also consider the IMEX-CNAB (Crank-Nicolson, Adams-Bashforth) scheme

$$\boldsymbol{v}_{m+1} - \boldsymbol{v}_m = \Delta\tau\lambda(\boldsymbol{J} - c\boldsymbol{I})\frac{3\boldsymbol{v}_m - \boldsymbol{v}_{m-1}}{2} + \Delta\tau(\boldsymbol{D} - \lambda(1-c)\boldsymbol{I})\frac{\boldsymbol{v}_{m+1} + \boldsymbol{v}_m}{2}, \tag{9}$$

and the IMEX-BDF2 scheme

$$\begin{aligned} \frac{3}{2}\boldsymbol{v}_{m+1} - 2\boldsymbol{v}_m + \frac{1}{2}\boldsymbol{v}_{m-1} = \Delta\tau\lambda(\boldsymbol{J} - c\boldsymbol{I})(2\boldsymbol{v}_m - \boldsymbol{v}_{m-1}) \\ +\Delta\tau(\boldsymbol{D} - \lambda(1-c)\boldsymbol{I})\boldsymbol{v}_{m+1}. \end{aligned} \tag{10}$$

These schemes were previously studied in a more general context in [2,10], for example. Above, we have included the extra parameter $c \in [0,1]$ for the convex combination of the zeroth-order term λv between the explicit and implicit part. This leads to a family of methods for each scheme, with a particular method defined by the value of c. We adopt the finite difference method for the spatial discretization; see [19] for details on the discretization. We note, however, that a different spatial discretization method could be chosen as well, for instance, the finite element method.

The considered schemes are constructed by starting from an implicit method, which is then combined with an extrapolation procedure for the explicit part. Schemes of this type are known to be consistent [11,13]. Also, the order of an IMEX linear multistep method is the minimum order of the explicit and implicit method considered separately [13]. Therefore, the IMEX methods considered here are of order 2.

The stability of a numerical scheme is usually studied under Fourier stability analysis. A finite difference scheme is stable if the errors made at one time step do not magnify as the computations are carried forward in time. A scheme that can be shown to be stable in the limit $\Delta\tau \to 0$ is called a (zero) stable scheme. A solution produced by such a scheme is guaranteed to remain bounded under finite time. In practice, however, time step sizes are of fixed size, and for optimal computational efficiency

one usually wants to choose the largest possible time step to obtain a desired level of accuracy. Zero stability is a necessary condition for the good behavior of the scheme, but for practical computations it is not sufficient. The actual required time step size for stability can be especially restrictive for stiff problems or schemes with more than two time levels. In the worst case, a scheme can be zero stable, but at the same time *unstable* for any fixed $\Delta\tau > 0$.
A theoretical definition for stability that is also sufficient for practical purposes is that of absolute stability. A scheme is absolutely stable if it can be shown to be stable for a range of $\Delta\tau$. This is a more stringent requirement, and zero stability clearly follows from absolute stability. Unfortunately, absolute stability of a scheme is often difficult to prove. The usual approach is to apply the scheme to the linear test problem and analyze its stability. In the case of IMEX-schemes, the linear test equation reads

$$v'(\tau) = \nu_B v(\tau) + \nu_C v(\tau), \tag{11}$$

where ν_B and ν_C are the complex eigenvalues of the explicit and implicit part of the scheme, respectively.
By applying the so-called method of lines approach on the PIDE (6), we obtain a semi-discrete linear system of ODEs

$$\boldsymbol{v}'(\tau) = \boldsymbol{D}\boldsymbol{v}(\tau) + \boldsymbol{J}\boldsymbol{v}(\tau) - \lambda\boldsymbol{v}(\tau), \quad \tau \geq 0. \tag{12}$$

Stability results for the test equation (11) can be readily extended to linear systems with commuting matrices. The stability of IMEX-schemes in a commutative framework was discussed in [10], for example. However, above $\boldsymbol{D}$ and $\boldsymbol{J}$ do not commute in general. Nevertheless, in many practical applications it has been found that time step size restrictions based on the linear test equation are accurate even in the noncommutative case [12].
First, we will analyze the stability of the explicit part in the IMEX-schemes (8), (9) and (10). In this special case the formal connection between the test equation (11) and the system of ODEs (12) is maintained. This enables us to present some theoretical stability bounds for the PIDE (2) in the special case $\boldsymbol{D} = \boldsymbol{0}$.

3.1 Explicit stability

The stability of an IMEX linear multistep method, as in [10], is determined by the roots of the characteristic equation

$$\sum_{i=0}^{k} a_i \zeta^{k-1} = \nu_B \sum_{i=1}^{k} b_i \zeta^{k-1} + \nu_C \sum_{i=0}^{k} c_i \zeta^{k-1} = 0. \tag{13}$$

The scheme is stable if all roots satisfy $|\zeta| \leq 1$, with strict inequality for multiple roots. We adopt the formulation in [10] by dividing the equation by $|\zeta^k|$ and substituting $z = 1/\zeta$. Now the characteristic equation reads

$$A(z) = \nu_B B(z) + \nu_C C(z), \tag{14}$$

where A, B and C are given by

$$A(z) = \sum_{i=0}^{k} a_i z^i, \quad B(z) = \sum_{i=1}^{k} b_i z^i, \quad C(z) = \sum_{i=0}^{k} c_i z^i. \tag{15}$$

Stability holds if all roots satisfy $|z| \geq 1$, again with strict inequality if $|z|$ is a multiple root. A necessary condition for this is

$$A(z) - \nu_B B(z) - \nu_C C(z) \neq 0, \quad \text{for all} \quad |z| < 1. \tag{16}$$

Apart from the possibility of multiple roots of modulus 1 this is also a sufficient condition.
For the IMEX-midpoint scheme (8) the polynomials in (14) are given by

$$A(z) = \frac{1}{2}(1 - z^2), \qquad B(z) = z, \qquad C(z) = \frac{1}{2}(1 + z^2), \tag{17}$$

and ν_B and ν_C are the eigenvalues of

$$\Delta\tau\lambda(\boldsymbol{J} - c\boldsymbol{I})\boldsymbol{v} = \nu_B \boldsymbol{v} \quad \text{and} \quad \Delta\tau(\boldsymbol{D} - (1-c)\lambda\boldsymbol{I})\boldsymbol{v} = \nu_C \boldsymbol{v}. \tag{18}$$

Let the eigenvalues $\nu_{\boldsymbol{J}}$ and $\nu_{\boldsymbol{D}}$ be given by

$$\boldsymbol{J}\boldsymbol{v} = \nu_{\boldsymbol{J}}\boldsymbol{v}, \quad \text{and} \quad \boldsymbol{D}\boldsymbol{v} = \nu_{\boldsymbol{D}}\boldsymbol{v}. \tag{19}$$

Then it holds

$$\nu_B = \Delta\tau\lambda(\nu_{\boldsymbol{J}} - c) \quad \text{and} \quad \nu_C = \Delta\tau(\nu_{\boldsymbol{D}} - (1-c)\lambda). \tag{20}$$

Let us consider the special case that $\nu_{\boldsymbol{D}} = 0$. Then

$$\nu_C = -\Delta\tau\lambda(1-c). \tag{21}$$

Substituting ν_B in (20) and ν_C in (21) into the characteristic polynomial gives

$$\frac{1}{2}(1 - z^2) = \Delta\tau\lambda(\nu_{\boldsymbol{J}} - c)z - \frac{1}{2}\Delta\tau\lambda(1-c)(1+z^2). \tag{22}$$

Solving $\nu_{\boldsymbol{J}}$ from the above equation yields

$$\nu_{\boldsymbol{J}} = c + \frac{1}{2z}\left[\frac{1}{\Delta\tau\lambda}(1-z^2) + (1-c)(1+z^2)\right]. \tag{23}$$

The boundary of the stability region for $\nu_{\boldsymbol{J}}$ is obtained by setting $z = e^{i\theta}$, substituting this into (23), and letting θ vary in the range $[0, 2\pi]$. The expression for the stability boundary simplifies to

$$\nu_{\boldsymbol{J}}(\theta) = c + (1-c)\cos\theta - \frac{i}{\lambda\Delta\tau}\sin\theta. \tag{24}$$

This defines an ellipse centered at c, the radius along the real axis being $1-c$, and the radius along the imaginary axis being $\frac{1}{\lambda\Delta\tau}$. Stability boundaries for the IMEX-midpoint scheme under various $\lambda\Delta\tau$ and c are plotted in Figure 1. The gray unit ball illustrates the area of possible eigenvalues $\nu_{\boldsymbol{J}}$. This suggests that only for the choice $c=0$ the IMEX-midpoint scheme is stable for all possible $\nu_{\boldsymbol{J}}$. In [15, 16] the authors propose a similar scheme as $c=1$ here, with the difference that they include the term rv in the explicit part. They show that the scheme is zero stable. Our analysis suggests, however, that this type of a scheme can be particularly ill-suited for practical computations because the scheme may be unstable for any $\lambda\Delta\tau > 0$. The polynomials for the IMEX-CNAB scheme (9) are given by

$$A(z) = (1-z), \qquad B(z) = \frac{1}{2}(3z - z^2), \qquad C(z) = \frac{1}{2}(1+z). \tag{25}$$

Again for $\nu_{\boldsymbol{J}}$ this leads to

$$\nu_{\boldsymbol{J}} = c + \frac{2}{3z - z^2}\left[\frac{1}{\Delta\tau\lambda}(1-z) + \frac{1}{2}(1-c)(1+z)\right]. \tag{26}$$

We have chosen the IMEX-CNAB scheme to represent methods from the IMEX-Adams family of methods; see Example 2.3 in [10]. We also briefly considered the other IMEX-Adams family schemes advocated by Nevanlinna and Liniger in [18] and Ascher et al. in [2], but we omit these results since they produced roughly similar results. Stability boundaries for the IMEX-CNAB scheme are plotted in Figure 2. The polynomials for the IMEX-BDF2 scheme (10) are given by

$$A(z) = \frac{1}{2}(3-z)(1-z), \qquad B(z) = z(2-z), \qquad C(z) = 1. \tag{27}$$

Similarly as above, we get for $\nu_{\boldsymbol{J}}$

$$\nu_{\boldsymbol{J}} = c + \frac{1}{z(2-z)}\left[\frac{1}{2\Delta\tau\lambda}(3-z)(1-z) + (1-c)\right]. \tag{28}$$

The stability boundaries for this scheme are plotted in Figure 3.

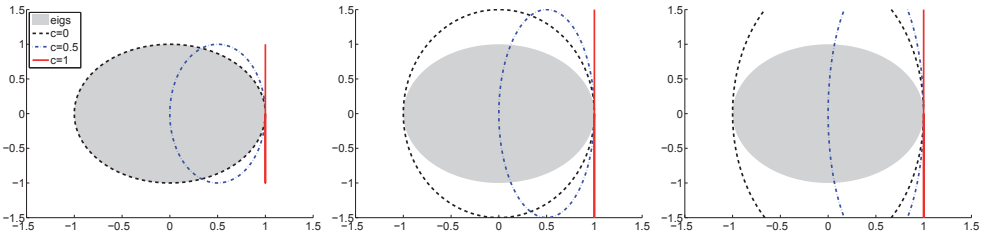

Figure 1: Stability boundaries for $c = \{0, \frac{1}{2}, 1\}$ under the IMEX-midpoint scheme with $\nu_{\boldsymbol{D}} = 0$ and $\lambda\Delta\tau = 1$ (left), $\lambda\Delta\tau = \frac{2}{3}$ (center) and $\lambda\Delta\tau = \frac{1}{2}$ (right).

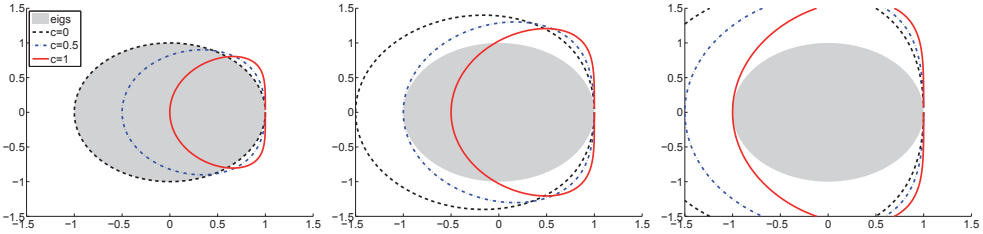

Figure 2: Stability boundaries for $c = \{0, \frac{1}{2}, 1\}$ under the IMEX-CNAB scheme with $\nu_{\boldsymbol{D}} = 0$ and $\lambda\Delta\tau = 1$ (left), $\lambda\Delta\tau = \frac{2}{3}$ (center) and $\lambda\Delta\tau = \frac{1}{2}$ (right).

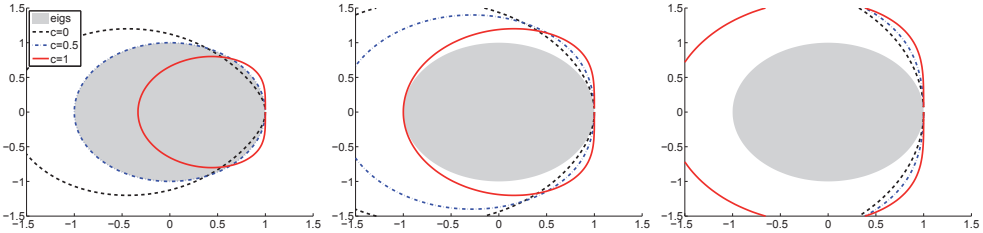

Figure 3: Stability boundaries for $c = \{0, \frac{1}{2}, 1\}$ under the IMEX-BDF2 scheme with $\nu_{\boldsymbol{D}} = 0$ and $\lambda\Delta\tau = 1$ (left), $\lambda\Delta\tau = \frac{2}{3}$ (center) and $\lambda\Delta\tau = \frac{1}{2}$ (right).

We will now formally investigate the explicit stability regions of the IMEX schemes. We will assume that $\boldsymbol{J}$ does not have error due to the truncation of the domain. This leads to a non-negative right stochastic matrix $\boldsymbol{J}$. An alternative path is to assume that $\boldsymbol{J}$ has truncation error and is non-negative. In this case $|\nu_{\boldsymbol{J}}| < 1$ holds due to the Gershgorin circle theorem. Thus, roots of modulus 1 do not exist, and the theoretical results below would be valid even if $\boldsymbol{J}$ is not right stochastic. We assume that $\boldsymbol{J}$ does not have truncation error because it is a more general framework.
In the following analysis, we need the following lemma.

Lemma 1. *If $\boldsymbol{J}$ is strictly positive, then $|\nu_{\boldsymbol{J}}| \leq 1$ and roots of modulus 1 are simple.*

Proof. The Gershgorin circle theorem states that the eigenvalues

$$\lambda(\boldsymbol{A}) \in \bigcup_{1 \leq i \leq N} D(a_{ii}, \sum_{j \neq i} |a_{ij}|), \tag{29}$$

where a_{ij} are elements of a square matrix $\boldsymbol{A}$, and $D(a, r)$ is a disc of radius r centered at a. Since $\boldsymbol{J}$ is a right stochastic matrix, the theorem applied to $\boldsymbol{J}$ gives $\lambda(\boldsymbol{J}) \in D(0, 1)$. That is, the eigenvalues of $\boldsymbol{J}$ are bounded by $|\nu_{\boldsymbol{J}}| \leq 1$. Since $\boldsymbol{J}$ is strictly positive, the Perron-Frobenius theorem states that 1 is a Perron-Frobenius eigenvalue. The roots corresponding to the Perron-Frobenius eigenvalue are simple. □

Proposition 1. *If $\boldsymbol{J}$ is strictly positive and $\nu_{\boldsymbol{D}} = 0$, then the IMEX-midpoint scheme is absolutely stable for $\lambda\Delta\tau < 1$ and $c = 0$.*

Proof. The following shows that under the assumptions $|\nu_J| \leq 1$ and $\lambda\Delta\tau < 1$, $z > 1$ has to hold. From $|\nu_J| \leq 1$, it follows

$$4|\nu_J|^2 = 4\left|\frac{1}{z}\left[\frac{1}{2\lambda\Delta\tau}(1-z^2) + \frac{1}{2}(1+z^2)\right]\right|^2 \leq 4. \tag{30}$$

From this it follows

$$\left|\frac{1}{\lambda\Delta\tau}(1-z^2) + 1 + z^2\right|^2 \leq |z|^2. \tag{31}$$

Denote $d = \frac{1}{\lambda\Delta\tau} - 1$ and, thus, $\frac{1}{\lambda\Delta\tau} = d + 1$. Note that $d > 0$ holds due to the assumption $\lambda\Delta\tau < 1$. The above is

$$\left|(d+1)(1-z^2) + 1 + z^2\right| = \left|2 + d(1-z^2)\right| \leq |z|^2. \tag{32}$$

Let $z = a + bi$ for real a and b. The above is

$$|2 + d(1 - a^2 + b^2 - 2abi)|^2 = [2 + d(1 - a^2 + b^2)]^2 + d^2[2ab]^2 \leq 4(a^2 + b^2) \tag{33}$$

which leads to

$$4 - 4(a^2 + b^2) + 4d(1 - a^2 + b^2) + d^2(1 - a^2 + b^2)^2 + 4d^2a^2b^2 \leq 0. \tag{34}$$

The following shows that $|z|^2 = a^2 + b^2 < 1$ leads to contradiction and, thus, $|z| \geq 1$ has to hold. From the above it follows

$$4d(1 - a^2 + b^2) + d^2(1 - a^2 + b^2)^2 + 4d^2a^2b^2 \leq 0. \tag{35}$$

Dividing by $d > 0$ leads to

$$4(1 - a^2 + b^2) + d(1 - a^2 + b^2)^2 + 4da^2b^2 \leq 0. \tag{36}$$

As $a^2b^2 \geq 0$, it holds

$$4(1 - a^2 + b^2) + d(1 - a^2 + b^2)^2 \leq 0. \tag{37}$$

It is easy to see that $1 - a^2 + b^2 > 0$ holds for a and b such that $a^2 + b^2 < 1$. Therefore,

$$4(1 - a^2 + b^2) + d(1 - a^2 + b^2)^2 > 0 \tag{38}$$

which is a contradiction with the previous inequality. □

Notice that we assumed that $\boldsymbol{J}$ is strictly positive. This assumption can be relaxed to non-negative $\boldsymbol{J}$, but then one has to additionally assume that possible roots at real $z = \pm 1$ are simple.

Proposition 2. *If $\boldsymbol{J}$ is strictly positive then the IMEX-BDF2 scheme with $\nu_{\boldsymbol{D}} = 0$ is absolutely stable for all $\lambda\Delta\tau < \frac{2}{3}$ and $c \in [0, 1]$.*

Proof. We know that the characteristic equation (14) is satisfied for

$$\nu_{\boldsymbol{J}} = c + \frac{1}{z(2-z)}\left[\frac{1}{2\Delta\tau\lambda}(3-z)(1-z) + (1-c)\right]. \tag{39}$$

From Lemma 1 it follows that $|\nu_{\boldsymbol{J}}| \leq 1$. Assume that $|z| < 1$ and that equation (39) holds. In the following we will show that this leads to a contradiction. Denote $k = \frac{1}{2\lambda\Delta\tau} > \frac{3}{4}$ and $z = a + ib$. Multiplying (39) by z yields

$$\begin{aligned}
\nu_{\boldsymbol{J}} z &= cz + \frac{1}{(2-z)}[k(3-z)(1-z) + (1-c)] \\
&= cz + k\left(2 + \frac{1}{z-2} - z\right) + \frac{1-c}{2-z} \\
&= cz + k(2-z) + \frac{1-c-k}{2-z} \\
&= cz + k(2-z) + \frac{(1-c-k)(2-\overline{z})}{(2-a)^2 + b^2} \\
&= ca + icb + k(2-a-ib) + \frac{(1-c-k)(2-a+ib)}{(2-a)^2+b^2}.
\end{aligned} \tag{40}$$

Now we can show that $\Re(\nu_{\boldsymbol{J}} z) \geq 1$, which leads to $|\nu_{\boldsymbol{J}} z| \geq 1$.

$$\begin{aligned}
\Re(\nu_{\boldsymbol{J}} z) &= ca + k(2-a) + \frac{(1-c-k)(2-a)}{(2-a)^2+b^2} \\
&= ca + \frac{(1-c)(2-a)}{(2-a)^2+b^2} + k(2-a)\left[1 - \frac{1}{(2-a)^2+b^2}\right].
\end{aligned} \tag{41}$$

In the above expression, the coefficient of k is positive. Thus, the smallest possible value of $k > \frac{3}{4}$ minimizes the expression and the inequality holds

$$\begin{aligned}
\Re(\nu_{\boldsymbol{J}} z) &= ca + \frac{(1-c)(2-a)}{(2-a)^2+b^2} + k(2-a)\left[1 - \frac{1}{(2-a)^2+b^2}\right] \\
&> ca + \frac{(1-c)(2-a)}{(2-a)^2+b^2} + \frac{3}{4}(2-a)\left[1 - \frac{1}{(2-a)^2+b^2}\right] \\
&= ca + \frac{(1-c)(2-a) - \frac{3}{4}(2-a)}{(2-a)^2+b^2} + \frac{3}{4}(2-a) \\
&= ca + \frac{1}{4}(2-a)\frac{1-4c}{(2-a)^2+b^2} + \frac{3}{4}(2-a).
\end{aligned} \tag{42}$$

In the above expression, for the coefficient of c it holds

$$a - \frac{2-a}{(2-a)^2+b^2} \leq a - \frac{2-a}{(2-a)^2} = a - \frac{1}{2-a} \leq 0. \tag{43}$$

Thus, the expression is minimized by the choice $c = 1$ and the next inequalities hold

$$\begin{aligned}\Re(\nu_{\boldsymbol{J}} z) &> ca + \frac{1}{4}(2-a)\frac{1-4c}{(2-a)^2+b^2} + \frac{3}{4}(2-a) \\ &\geq a - \frac{3}{4}\frac{2-a}{(2-a)^2+b^2} + \frac{3}{4}(2-a) \\ &\geq a - \frac{3}{4}\frac{2-a}{(2-a)^2} + \frac{3}{4}(2-a) \\ &= \frac{1}{4}\left[a + 6 - \frac{3}{2-a}\right]\end{aligned} \tag{44}$$

The function $\frac{1}{4}\left[a+6-\frac{3}{2-a}\right]$ first increases from 1 at $a = -1$ to $\frac{1}{2}\left[4-\sqrt{3}\right] > 1$ at $a = 2-\sqrt{3}$, and then it decreases to 1 at $a = 1$. The above analysis shows that $\Re(\nu_{\boldsymbol{J}} z) > 1$ and, thus, $|\nu_{\boldsymbol{J}} z| > 1$. This leads to $|\nu_{\boldsymbol{J}}||z| = |\nu_{\boldsymbol{J}} z| \geq 1$, and also $|\nu_{\boldsymbol{J}}| \geq |\frac{1}{z}| > 1$, which is a contradiction with Lemma 1. Hence, no $|z| < 1$ exists such that (39) holds, which means that the necessary condition for stability (16) holds. This is also sufficient, since Lemma 1 guarantees that roots of modulus 1 are simple. □

The following result can be proven in the same way as Proposition 2.

Proposition 3. *If $\boldsymbol{J}$ is strictly positive then the IMEX-CNAB scheme with $\nu_{\boldsymbol{D}} = 0$ is absolutely stable for all $\lambda\Delta\tau < \frac{1}{2}$ and $c \in [0,1]$.*

Remark 1. *Above, we let $c \in [0,1]$. It can also be shown that the IMEX-BDF2 scheme is stable for all $\lambda\Delta\tau < 1$ and $c = \frac{1}{2}$. This is illustrated in Figure 3 (left). Also, it can be shown that the IMEX-CNAB scheme is stable for all $\lambda\Delta\tau < 1$ and $c = 0$, see Figure 2 (left).*

Under Merton and Kou models $\boldsymbol{J}$ is strictly positive for any quadrature with positive weights. The following corollaries result from Proposition 1, 2 and 3.

Corollary 1. *The IMEX-midpoint scheme with $\nu_{\boldsymbol{D}} = 0$ is absolutely stable under Kou and Merton models for $\lambda\Delta\tau < 1$ and $c = 0$.*

Corollary 2. *The IMEX-BDF2 scheme with $\nu_{\boldsymbol{D}} = 0$ is absolutely stable under Kou and Merton models for all $\lambda\Delta\tau < \frac{2}{3}$ and $c \in [0,1]$.*

Corollary 3. *The IMEX-CNAB scheme with $\nu_{\boldsymbol{D}} = 0$ is absolutely stable under Kou and Merton models for all $\lambda\Delta\tau < \frac{1}{2}$ and $c \in [0,1]$.*

3.2 Implicit stability

While the theoretical connection between the linear test problem (11) and the system of ODEs (12) does not hold in the full generality of the jump-diffusion model ($\nu_{\boldsymbol{J}} \neq 0$ and $\nu_{\boldsymbol{D}} \neq 0$), we can nevertheless try to predict the behavior of the IMEX-schemes from the stability boundaries of the linear test problem. There is no guarantee

that they are accurate for the system of ODEs (12), but this is generally the best prediction that can be done.
A sufficient condition for the eigenvalue $\nu_{\boldsymbol{D}}$ being real and nonpositive is, for example, that $-\boldsymbol{D}$ is tridiagonal and M-matrix, as in this case $\boldsymbol{D}$ is a quasi-symmetric tridiagonal matrix. Under $r \geq 0$, the M-matrix property can be attained by adding artificial diffusion into the model, as was done in [22] for example. Using the central difference discretizations for v_{xx} leads the lower bound for $\nu_{\boldsymbol{D}}$ being of order $-2\sigma^2 X^2/(\Delta x)^2$, where X is the right end of the computational domain and Δx is the grid step size.
By repeating the procedure as before, but this time with $\nu_{\boldsymbol{D}} \neq 0$, we obtain the following expression for $\nu_{\boldsymbol{J}}$ under the IMEX-midpoint scheme, for example.

$$\nu_{\boldsymbol{J}} = c + \frac{1}{2z}\left[\frac{1}{\Delta\tau\lambda}(1-z^2) + (1-c-\frac{\nu_{\boldsymbol{D}}}{\lambda})(1+z^2)\right]. \tag{45}$$

The stability region is an ellipse similar to the case $\nu_{\boldsymbol{D}} = 0$ defined in (24), but this time with a radius along the real axis given by $1 - c - \nu_{\boldsymbol{D}}/\lambda$. This is illustrated in Figure 4. The stability region in (45) improves for any real $\nu_{\boldsymbol{D}} < 0$. Thus, under these assumptions, the stability of the scheme only improves with the implicit part included. The same is true for the IMEX-CNAB and the IMEX-BDF2 schemes, which is illustrated in Figures 5 and 6. This suggests that under suitable assumptions, the special case $\nu_{\boldsymbol{D}} = 0$ can be considered as the worst case stability scenario. While this formally applies only to the test problem, it is possible to test the practical performance of the methods applied to the PIDE (6) through numerical experiments.

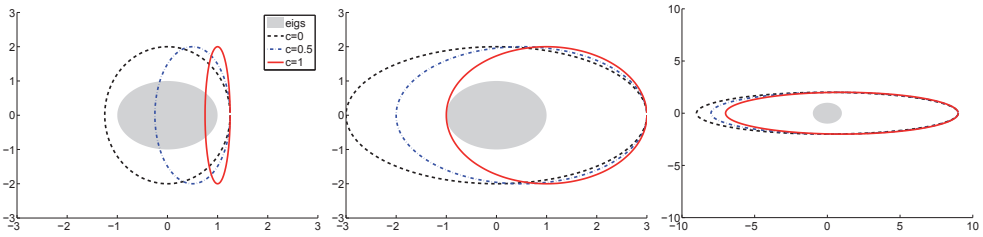

Figure 4: Stability boundaries for $c = \{0, \frac{1}{2}, 1\}$ under the IMEX-midpoint scheme with $\lambda = \frac{1}{2}$, $\Delta\tau = 1$ and $\nu_{\boldsymbol{D}} = -\frac{1}{8}$ (left), $\nu_{\boldsymbol{D}} = -1$ (center) and $\nu_{\boldsymbol{D}} = -4$ (right).

4 Numerical Experiments

In this section we present numerical results computed with the IMEX-midpoint, IMEX-CNAB and IMEX-BDF2 schemes. We price European and American put options under the parameters listed in Table 1. A uniform grid with N spatial and M temporal nodes is used, and the spatial truncation boundary is set to $X = 400$. We take $N = 6401$, which should be a sufficiently fine spatial grid to accurately observe

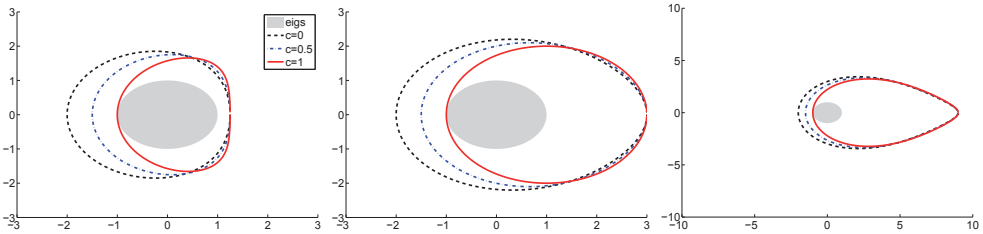

Figure 5: Stability boundaries for $c = \{0, \frac{1}{2}, 1\}$ under the IMEX-CNAB scheme with $\lambda = \frac{1}{2}$, $\Delta\tau = 1$ and $\nu_{\boldsymbol{D}} = -\frac{1}{8}$ (left), $\nu_{\boldsymbol{D}} = -1$ (center) and $\nu_{\boldsymbol{D}} = -4$ (right).

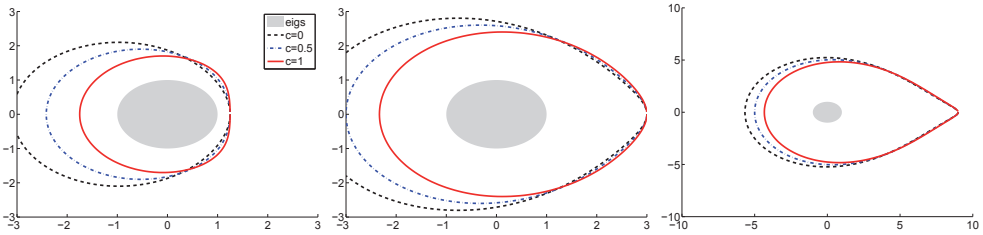

Figure 6: Stability boundaries for $c = \{0, \frac{1}{2}, 1\}$ under the IMEX-BDF2 scheme with $\lambda = \frac{1}{2}$, $\Delta\tau = 1$ and $\nu_{\boldsymbol{D}} = -\frac{1}{8}$ (left), $\nu_{\boldsymbol{D}} = -1$ (center) and $\nu_{\boldsymbol{D}} = -4$ (right).

the convergence in the temporal dimension. The first two time steps are taken with the first-order accurate one-step IMEX-Euler scheme. The reference prices were computed numerically with $N = 6401, M = 16384$ using a fixed-point iteration for European options [21] and for American options [19]. The amount of spatial nodes is kept constant, while the number of time steps is doubled in each refinement. The computations were performed on a PC with a 2.3 GHz Intel Xeon E5410 processor. The root mean square errors (RMSE) were computed at spatial nodes $x_i = \{90, 100, 110\}$. More precisely, RMSE $= \sqrt{((v_1 - v_1^*)^2 + (v_2 - v_2^*)^2 + (v_3 - v_3^*)^2)/3}$, where v_i^* is the reference price and v_i is the computed price at $x_i = \{90, 100, 110\}$. The a priori chosen convex combination parameters are $c = \{0, 0.5, 1\}$.

Expiry Time T	1
Strike K	100
Risk free rate r	0.05
Volatility σ	0.15
Jump arrival rate λ	$\{0.5, 5, 50\}$
Jump model type	Kou
Jump distribution parameters	$\alpha_1 = 3, \alpha_2 = 3, p = 0.333333$

Table 1: Parameter setup for European and American put options

First, we price European options under the Kou model with the parameters listed in Table 1. The temporal errors for $\lambda = \{0.5, 5, 50\}$ are plotted in Figures 7, 8, and 9 for the IMEX-midpoint, IMEX-CNAB, and IMEX-BDF2 schemes, respectively. Next, we price American options under the Kou model with the same parameters listed in Table 1. The temporal errors for $\lambda = \{0.5, 5, 50\}$ are plotted in Figures 10, 11, and 12 for the IMEX-midpoint, IMEX-CNAB, and IMEX-BDF2 schemes, respectively. The IMEX-midpoint scheme with the zeroth-order term treated explicitly ($c = 1$) becomes rapidly unstable with higher values of λ. The accuracy disparity between different values of c is insignificant with small to moderate values of λ, but becomes evident with larger values of λ.

Finally, we compare the IMEX-midpoint ($c = 0$), IMEX-CNAB ($c = 0$) and IMEX-BDF2 ($c = 0$) schemes against each other, and against an alternative method, namely, the fixed-point iteration that was used to compute the reference prices. This comparison is plotted in Figure 13 for European options and in Figure 14 for American options for $\lambda = \{0.5, 5, 50\}$. This comparison is not fair in terms of computational effort, since each time step takes significantly longer with the iterative method; see [20] for a comparison with CPU times. The comparison is nevertheless included to illustrate the general level of accuracy of the IMEX-schemes. This shows that while the IMEX-schemes are very fast and second-order convergent, their (initial) accuracy deteriorates with higher values of λ. In other words, for high λ it takes an IMEX scheme more time steps to reach the same level of accuracy as the iterative methods. However, also the iterative methods require more iterations per time step with higher values of λ; see Corollary 1 in [19].

The IMEX schemes are about twice as fast with a small λ, see [20] for example, and several times faster with a large λ than the iterative methods. Furthermore, Figures 13 and 14 indicate that the accuracy disparity between the IMEX-schemes and iterative methods is negligible with low to moderate values of λ. Thus, for low to moderate λ the IMEX schemes are clearly more efficient. In most cases the IMEX-CNAB $c = 0$ scheme produced the smallest error among the considered IMEX schemes in our numerical experiments.

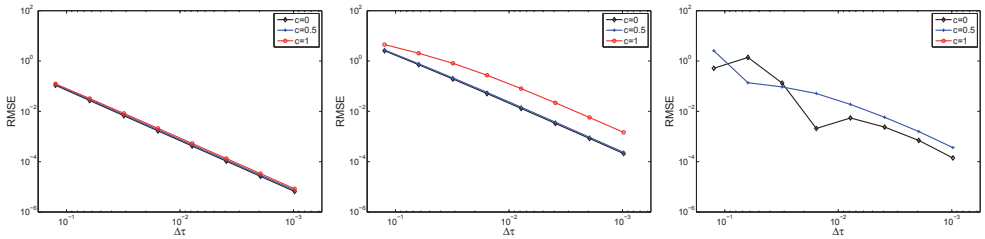

Figure 7: Temporal errors for the European put option under the IMEX-midpoint scheme with $\lambda = 0.5$ (left), $\lambda = 5$ (center) and $\lambda = 50$ (right, $c = 1$ unstable and is off the chart).

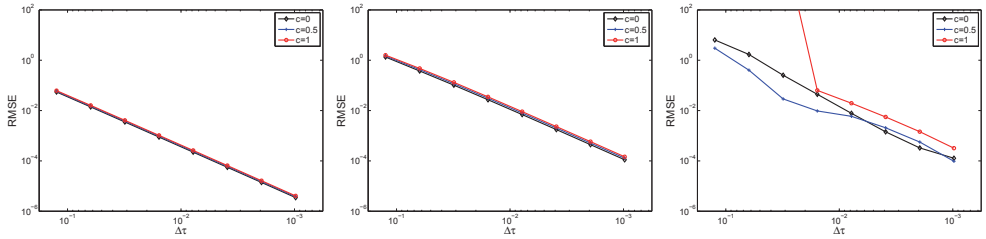

Figure 8: Temporal errors for the European put option under the IMEX-CNAB scheme with $\lambda = 0.5$ (left), $\lambda = 5$ (center) and $\lambda = 50$ (right).

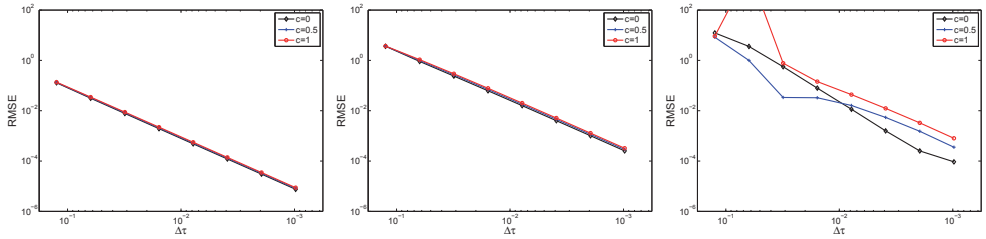

Figure 9: Temporal errors for the European put option under the IMEX-BDF2 scheme with $\lambda = 0.5$ (left), $\lambda = 5$ (center) and $\lambda = 50$ (right).

5 Conclusions

We considered the accuracy and absolute stability of several IMEX schemes applied to the PIDE (6) with varying convex combinations of the zeroth-order term λv. The only stable choice for the IMEX-midpoint scheme is the fully implicit discretization of λv with the convex combination parameter $c = 0$. In particular, the choice $c = 1$ proved to be very unstable for this scheme. A scheme similar to $c = 1$ was proposed in [15, 16], for example. We find that this type of a scheme is not suitable for larger values of λ . The IMEX-CNAB and IMEX-BDF2 schemes are stable for all $c \in [0, 1]$, with the IMEX-CNAB scheme having a slightly more restrictive stability condition

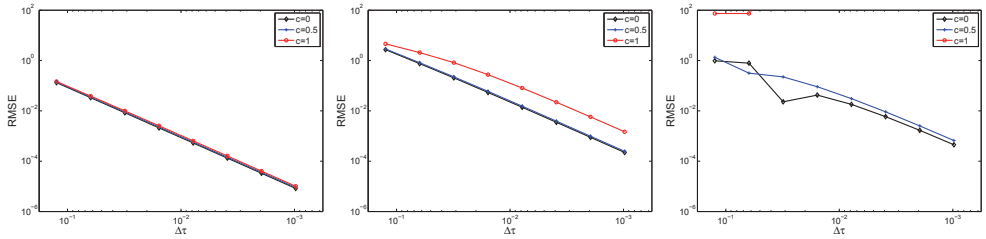

Figure 10: Temporal errors for the American put option under the IMEX-midpoint scheme with $\lambda = 0.5$ (left), $\lambda = 5$ (center) and $\lambda = 50$ (right).

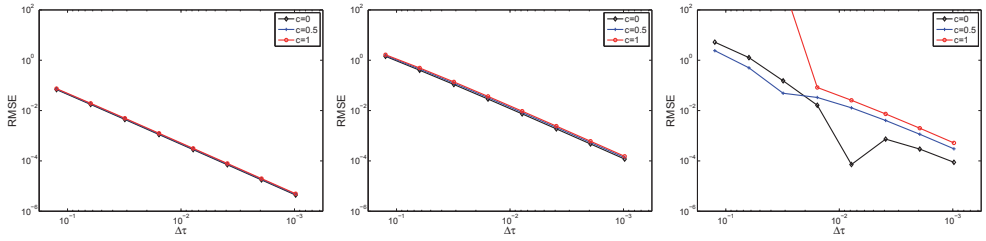

Figure 11: Temporal errors for the American put option under the IMEX-CNAB scheme with $\lambda = 0.5$ (left), $\lambda = 5$ (center) and $\lambda = 50$ (right).

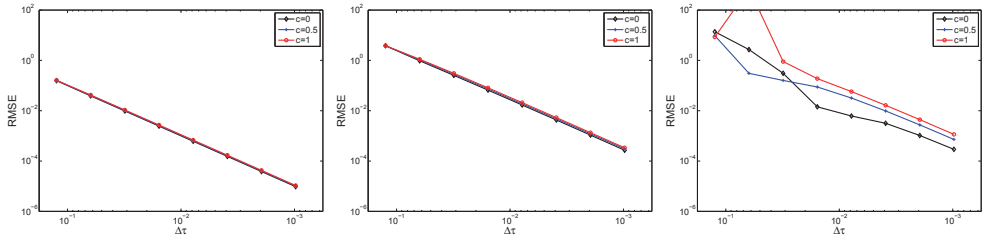

Figure 12: Temporal errors for the American put option under the IMEX-BDF2 scheme with $\lambda = 0.5$ (left), $\lambda = 5$ (center) and $\lambda = 50$ (right).

than the IMEX-BDF2 scheme.
The IMEX-CNAB scheme with $c = 0$ had the best accuracy among all considered IMEX schemes in our numerical experiments. We recommend the use of the IMEX-BDF2 scheme with $c = 0.5$ if maximal stability is crucial. Otherwise, we recommend the use of the IMEX-CNAB scheme with $c = 0$.

References

[1] L. Andersen and J. Andreasen, *Jump-diffusion processes: Volatility smile*

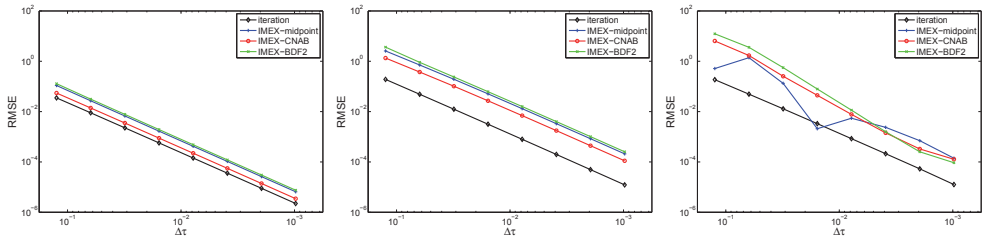

Figure 13: Comparison of temporal errors for the European put option under various methods with $\lambda = 0.5$ (left), $\lambda = 5$ (center) and $\lambda = 50$ (right).

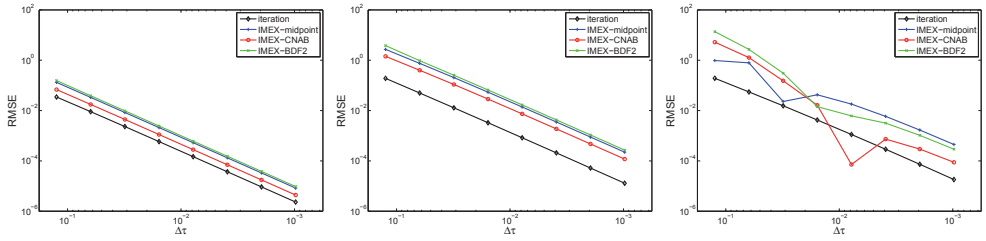

Figure 14: Comparison of temporal errors for the American put option under various methods with $\lambda = 0.5$ (left), $\lambda = 5$ (center) and $\lambda = 50$ (right).

fitting and numerical methods for option pricing, Rev. Deriv. Res., 4 (2000), pp. 231–262.

[2] U. Ascher, S. Ruuth, and B. Wetton, *Implicit-explicit methods for time-dependent partial differential equations*, SIAM Journal on Numerical Analysis, 32 (1995), pp. 797–823.

[3] M. Briani, R. Natalini, and G. Russo, *Implicit-explicit numerical schemes for jump-diffusion processes*, Calcolo, 44 (2007), pp. 33–57.

[4] P. Carr, H. Geman, D. B. Madan, and M. Yor, *The fine structure of asset returns: an empirical investigation*, J. Business, 75 (2002), pp. 305–332.

[5] R. Cont and P. Tankov, *Financial modelling with jump processes*, Chapman & Hall/CRC, Boca Raton, FL, 2004.

[6] R. Cont and E. Voltchkova, *A finite difference scheme for option pricing in jump diffusion and exponential Lévy models*, SIAM J. Numer. Anal., 43 (2005), pp. 1596–1626.

[7] Y. d'Halluin, P. A. Forsyth, and G. Labahn, *A penalty method for American options with jump diffusion processes*, Numer. Math., 97 (2004), pp. 321–352.

[8] F. Fang and C. W. Oosterlee, *A novel pricing method for European options based on Fourier-cosine series expansions*, SIAM J. Sci. Comput., 31 (2008/09), pp. 826–848.

[9] L. Feng and V. Linetsky, *Pricing options in jump-diffusion models: an extrapolation approach*, Oper. Res., 56 (2008), pp. 304–325.

[10] J. Frank, W. Hundsdorfer, and J. Verwer, *On the stability of implicit-explicit linear multistep methods*, Appl. Numer. Math., 25 (1997), pp. 193–205.

[11] E. Hairer, S. Norsett, and G. Wanner, *Solving Ordinary Differential Equations. I: Nonstiff Problems*, Springer, 1993.

[12] W. Hundsdorfer and S. J. Ruuth, *IMEX extensions of linear multistep methods with general monotonicity and boundedness properties*, J. Comput. Phys., 225 (2007), pp. 2016–2042.

[13] W. Hundsdorfer and J. Verwer, *A note on splitting errors for advection-reaction equations*, Appl. Numer. Math., 18 (1995), pp. 191–199.

[14] S. G. Kou, *A jump-diffusion model for option pricing*, Management Sci., 48 (2002), pp. 1086–1101.

[15] Y. Kwon and Y. Lee, *A second-order finite difference method for option pricing under jump-diffusion models*, SIAM J. Numer. Anal., 49 (2011), pp. 2598–2617.

[16] ——, *A second-order tridiagonal method for American options under jump-diffusion models*, SIAM J. Sci. Comput., 33 (2011), pp. 1860–1872.

[17] R. C. Merton, *Option pricing when underlying stock returns are discontinuous*, J. Financial Econ., 3 (1976), pp. 125–144.

[18] O. Nevanlinna and W. Liniger, *Contractive methods for stiff differential equations part II*, BIT Numerical Mathematics, 19 (1979), pp. 53–72.

[19] S. Salmi and J. Toivanen, *An iterative method for pricing American options under jump-diffusion models*, Appl. Numer. Math., 61 (2011), pp. 821–831.

[20] ——, *Comparison and survey of finite difference methods for pricing American options under finite activity jump-diffusion models*, International Journal of Computer Mathematics, 89 (2012), pp. 1112–1134.

[21] D. Tavella and C. Randall, *Pricing financial instruments: The finite difference method*, John Wiley & Sons, 2000.

[22] J. Toivanen, *Numerical valuation of European and American options under Kou's jump-diffusion model*, SIAM J. Sci. Comput., 30 (2008), pp. 1949–1970.

[23] P. Wilmott, *Derivatives*, John Wiley & Sons Ltd., Chichester, 1998.

PIV

ITERATIVE METHODS FOR PRICING AMERICAN OPTIONS UNDER THE BATES MODEL

by

S. Salmi, J. Toivanen and L. von Sydow 2013

Proceedings of 2013 International Conference on Computational Science,
Procedia Computer Science Series, Volume 18 pp. 1136-1144.

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 00 (2013) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Science, ICCS 2013

Iterative Methods for Pricing American Options under the Bates Model

Santtu Salmi[a], Jari Toivanen[a,b,*], Lina von Sydow[c]

[a]*Department of Mathematical Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland*
[b]*Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305, USA*
[c]*Department of Information Technology, Uppsala University, Box 337, 751 05 Uppsala, Sweden*

Abstract

We consider the numerical pricing of American options under the Bates model which adds log-normally distributed jumps for the asset value to the Heston stochastic volatility model. A linear complementarity problem (LCP) is formulated where partial derivatives are discretized using finite differences and the integral resulting from the jumps is evaluated using simple quadrature. A rapidly converging fixed point iteration is described for the LCP, where each iterate requires the solution of an LCP. These are easily solved using a projected algebraic multigrid (PAMG) method. The numerical experiments demonstrate the efficiency of the proposed approach. Furthermore, they show that the PAMG method leads to better scalability than the projected SOR (PSOR) method when the discretization is refined.

Keywords: American option; Bates model; Finite difference method; Iterative method; Linear complementarity problem

1. Introduction

In this paper we consider the numerical pricing of American options. Such options can be exercised prior to the date of maturity which leads to a free-boundary problem. This is in contrast to European options that can only be exercised on the date of maturity leading to an easier problem to solve. Since trading of options has grown to a tremendous scale during the last decades the need for accurate and effective numerical option pricing methods is obvious. The most common options give the holder either the right to sell (put option) or buy (call option) the underlying asset for the strike price. A mathematical model to describe the behavior of the underlying asset is needed to compute the option price. Many such models of varying complexity exist. Typically, more complicated models reproduce more realistic paths of the underlying asset and are hence better to give accurate option prices but they also make the numerical pricing process more challenging. The most commonly used model is the Black-Scholes model [1], which assumes the value of the underlying asset to follow a geometric Brownian motion. In the Merton model [2] log-normally distributed jumps are added to the Black-Scholes model, while in the Kou model [3] the jumps are log-doubly-exponentially distributed. By making the volatility a stochastic quantity the Heston model is derived [4], while the Bates model [5] combines the Merton model with the Heston model by adding log-normally distributed jumps to the latter one. Finally, the correlated jump model [6] also lets the volatility jump in the Bates model.

*Corresponding author.
E-mail address: toivanen@stanford.edu.

One way to price options is to employ a Monte-Carlo type solver that simulates the behavior of the underlying asset using the model employed and then compute discounted mean values. Such methods are known to have nonfavorable convergence properties and the treatment of the early exercise feature is nontrivial which is why we use another approach here. We formulate a linear complementarity problem (LCP) for a partial (integro-) differential equation (P(I)DE) operator for the price, discretize the P(I)DE, and then solve the resulting LCPs. Several methods have been proposed for solving the resulting LCPs. The Brennan and Schwartz algorithm [7] is a direct method for pricing American options under the Black-Scholes model; see also [8]. Numerical methods for pricing under the Heston model have been developed in [9], [10], [11], [12], [13], [14], for example. The treatment of the jumps in the Merton and Kou models have been studied in [15], [16], [17], [18], [19], [20], for example. Pricing under the Bates model has been considered in [21], [22] and under the correlated jump model in [23].

In this paper, we price American call options under the Bates model. The spatial partial derivatives in the resulting partial integro-differential operator are discretized using a seven-point finite difference stencil and the integral term is discretized using a simple quadrature rule. The Rannacher scheme [24] is employed for the time stepping. We solve the resulting LCPs by employing a fixed point iteration described and analyzed in [25] where each iteration requires the solution of an LCP. These are solved using a projected multigrid (PAMG) method which was recently introduced in [26]. The numerical experiments demonstrate that the proposed method is orders of magnitude faster than the projected successive overrelaxation (PSOR) method.

The outline of the paper is the following. The Bates model and an LCP formulation for an American call option is described in Sect. 2. In Sect. 3 the discretization of the LCPs is introduced and the iterative method to solve them is proposed in Sect. 4. Numerical experiments are presented in Sect. 5 and conclusions are given in Sect. 6.

2. Option Pricing Model

Here, we consider the Bates model [5] that combines the Merton jump model [2] and the Heston stochastic volatility model [4]. It describes the behavior of the asset value s and its variance y by the coupled stochastic differential equations

$$\begin{aligned} ds &= (\mu - \lambda\xi)sdt + \sqrt{y}sdw_1 + (J-1)sdn, \\ dy &= \kappa(\theta - y)dt + \sigma\sqrt{y}dw_2. \end{aligned}$$

Here μ is the growth rate of the asset value, κ is the rate of reversion to the mean level of y, θ is the mean level of y, and σ is the volatility of the variance y. The two Wiener processes w_1 are w_2 have the correlation ρ. The Poisson arrival process n has the rate λ and the jump size J is taken from a distribution

$$f(J) = \frac{1}{\sqrt{2\pi}\delta J}\exp\left(-\frac{[\ln J - (\gamma - \delta^2/2)]^2}{2\delta^2}\right),$$

where γ and δ define the mean and variance of the jump. The mean jump ξ is given by $\xi = \exp(\gamma) - 1$.

For simplicity, from now on we assume that the market prices of the volatility and jump risks are zero. Applying the Feynman-Kac formula to the Bates model we arrive at the following PIDE

$$\begin{aligned} 0 &= \tfrac{\partial u}{\partial \tau} - \tfrac{1}{2}ys^2\tfrac{\partial^2 u}{\partial s^2} - \rho\sigma ys\tfrac{\partial^2 u}{\partial s\partial y} - \tfrac{1}{2}\sigma^2 y\tfrac{\partial^2 u}{\partial y^2} - (r - q - \lambda\xi)s\tfrac{\partial u}{\partial s} - \kappa(\theta - y)\tfrac{\partial u}{\partial y} + (r+\lambda)u - \lambda\int_0^\infty u(Js, y, \tau)f(J)dJ \\ &= \tfrac{\partial u}{\partial \tau} - a_{11}\tfrac{\partial^2 u}{\partial s^2} - a_{12}\tfrac{\partial^2 u}{\partial s\partial y} - a_{22}\tfrac{\partial^2 u}{\partial y^2} - a_1\tfrac{\partial u}{\partial s} - a_2\tfrac{\partial u}{\partial y} + (r+\lambda)u - \lambda\int_0^\infty u(Js, y, \tau)f(J)dJ =: Lu, \end{aligned} \tag{1}$$

where u is the price of a European option, $\tau = T - t$ is the time to expiry and q is the dividend yield. The initial condition for (1) is defined by

$$u = g(s, y),$$

where g is the payoff function which gives the value of option at the expiry. In the following, we consider only call options. A similar approach can be also applied for put options. The payoff function for a call option with the strike price K is

$$g(s, y) = \max(s - K, 0).$$

For the computations, the unbounded domain is truncated to

$$(s, y, \tau) \in (0, S) \times (0, Y) \times (0, T] \tag{2}$$

with sufficiently large S and Y.

The price u of an American option under the Bates model satisfies an LCP

$$\begin{cases} Lu \geq 0, \\ u \geq g, \\ (Lu)(u - g) = 0. \end{cases} \tag{3}$$

We impose the boundary conditions

$$\begin{cases} u(0, y, \tau) = g(0, y), \\ u(S, y, \tau) = g(S, y), \end{cases} \qquad y \in (0, Y),$$

$$u_y(s, Y, \tau) = 0, \quad s \in (0, S).$$

Beyond the boundary $s = S$, the price u is approximated to be the same as the payoff g, that is, $u(s, y, \tau) = g(s, y)$ for $s \geq S$. On the boundary $y = 0$, the LCP (3) holds and no additional boundary condition needs to be posed.

3. Discretization

We will compute approximate prices u on a space-time grid defined by the grid points (x_i, y_j, τ_k), $0 \leq i \leq m$, $0 \leq j \leq n$, $0 \leq k \leq l$. In space we use a uniform grid with grid steps $\Delta s = S/m$ in the s-direction and $\Delta y = Y/n$ in the y-direction. We start by introducing a semidiscrete approximation for

$$u(s_i, y_j, \tau), \quad 0 \leq i \leq m, \quad 0 \leq j \leq n.$$

For the non cross-derivatives in (1) we use standard second-order, centered finite difference approximations. In this paper, we assume that the correlation ρ is negative. Due to the cross-derivative, we use a seven-point finite difference stencil. A similar stencil has been described in [21], [22]. For a positive correlation ρ, a suitable seven-point stencil is given in [10], [11]. The cross-derivative is approximated by

$$\begin{aligned} \frac{\partial^2 u}{\partial s \partial y}(s_i, y_j, \tau) \approx \frac{1}{2\Delta s \Delta y} \Big(& 2u(s_i, y_j, \tau) - u(s_{i-1}, y_{j+1}, \tau) - u(s_{i+1}, y_{j-1}, \tau) \\ & + (\Delta s)^2 \frac{\partial^2 u}{\partial s^2}(s_i, y_j, \tau) + (\Delta y)^2 \frac{\partial^2 u}{\partial y^2}(x_i, y_j, \tau) \Big). \end{aligned} \tag{4}$$

Due to the additional derivative terms in (4), we define modified coefficients for $\frac{\partial^2 u}{\partial s^2}$ and $\frac{\partial^2 u}{\partial y^2}$ as

$$\tilde{a}_{11} = a_{11} + \frac{1}{2}\frac{\Delta s}{\Delta y} a_{12}, \quad \text{and} \quad \tilde{a}_{22} = a_{22} + \frac{1}{2}\frac{\Delta y}{\Delta s} a_{12}.$$

To avoid positive weights in the computational stencil when the convection dominates the diffusion, we add some artificial diffusion according to

$$\hat{a}_{11} = \min\left\{\tilde{a}_{11}, \ -\frac{1}{2} b_1 \Delta s, \ \frac{1}{2} b_1 \Delta s\right\} \qquad \text{and} \qquad \hat{a}_{22} = \min\left\{\tilde{a}_{22}, \ -\frac{1}{2} b_2 \Delta y, \ \frac{1}{2} b_2 \Delta y\right\}.$$

This is equivalent to using a combination of one-sided and central differences for the convection part. The resulting matrix is an M-matrix with nonpositive off-diagonals and positive diagonal. It is strictly diagonally dominant when $r + \lambda > 0$.

The integral term in (1) at each grid point s_i is denoted by I_i. We start by making the change of variable $J = e^z$, to obtain

$$I_i = \int_0^\infty u(Js_i, y, \tau) f(J) dJ = \int_{-\infty}^\infty u(e^z x_i, y, \tau) p(z) dz,$$

where p is the probability density function of the normal distribution with mean $\gamma - \delta^2/2$ and variance δ^2 given by

$$p(z) = \frac{1}{\sqrt{2\pi}\delta} \exp\left(-\frac{[z - (\gamma - \delta^2/2)]^2}{2\delta^2}\right).$$

Then we decompose I_i into one integral over the computational domain defined in (2) and one integral over the remainder of the interval. The first part is then divided on the spatial grid so that we get

$$I_i = \sum_{j=0}^{n-1} I_{i,j} + \int_{\ln s_n - \ln s_i}^\infty g(e^z s_i, y) p(z) dz, \tag{5}$$

where

$$I_{i,j} = \int_{\ln s_{j+1} - \ln s_i}^{\ln s_j - \ln s_i} u(e^z s_i, y, \tau) p(z) dz. \tag{6}$$

The price function $u(s, y, \tau)$ needs to be approximated between each grid point pair (s_i, s_{i+1}) in order to define approximate values of $I_{i,j}$. For this, we use a piecewise linear interpolation

$$u(s, y, \tau) \approx \frac{s_{i+1} - s}{s_{i+1} - s_i} u(s_i, y, \tau) + \frac{s - s_i}{s_{i+1} - s_i} u(s_{i+1}, y, \tau) \tag{7}$$

for $s \in [s_i, s_{i+1}]$. Using (7) in (6) we get

$$I_{i,j} \approx \frac{e^\gamma}{2}\left[\operatorname{erf}\left(\frac{s_{i,j+1} - \delta^2/2}{\delta\sqrt{2}}\right) - \operatorname{erf}\left(\frac{s_{i,j} - \delta^2/2}{\delta\sqrt{2}}\right)\right]\alpha_j x_i + \frac{1}{2}\left[\operatorname{erf}\left(\frac{s_{i,j+1} + \delta^2/2}{\delta\sqrt{2}}\right) - \operatorname{erf}\left(\frac{s_{i,j} + \delta^2/2}{\delta\sqrt{2}}\right)\right]\beta_j x_i,$$

where $\operatorname{erf}(\cdot)$ is the error function, $s_{i,j} = \ln s_j - \ln s_i - \gamma$,

$$\alpha_j = \frac{u(s_{j+1}, y, \tau) - u(s_j, y, \tau)}{s_{j+1} - s_j}, \quad \text{and} \quad \beta_j = \frac{u(s_j, y, \tau) s_{j+1} - u(s_{j+1}, y, \tau) s_j}{s_{j+1} - s_j}.$$

The spatial discretization leads to a semi-discrete LCP

$$\begin{cases} \boldsymbol{u}_\tau + \boldsymbol{A}\boldsymbol{u} + \boldsymbol{a} \geq \boldsymbol{0}, \\ \boldsymbol{u} \geq \boldsymbol{g}, \\ (\boldsymbol{u}_\tau + \boldsymbol{A}\boldsymbol{u} + \boldsymbol{a})^T (\boldsymbol{u} - \boldsymbol{g}) = 0, \end{cases}$$

where $\boldsymbol{A}$ is an $(m+1)(n+1) \times (m+1)(n+1)$ matrix, $\boldsymbol{a}$ is a vector resulting from the second term in (5), $\boldsymbol{u}$ and $\boldsymbol{g}$ are vectors containing the grid point values of the price u and the payoff g, respectively. In the above LCP, the inequalities hold componentwise.

For the temporal discretization we use the Rannacher scheme [24]; see also [27]. The first four time steps are performed with the implicit Euler method with the time step $\Delta\tau/2$, and then the rest of the time steps are performed with the Crank-Nicolson method with the time step $\Delta\tau$, where $\Delta\tau = T/(l-2)$. Thus, the time grid is defined by

$$\tau_k = \begin{cases} \frac{k}{2(l-2)} T, & k = 0, 1, 2, 3, \\ \frac{k-2}{l-2} T, & k = 4, 5, \ldots, l. \end{cases}$$

The purpose of a few Euler steps in the beginning of the time-stepping process is to damp oscillatory components of the solution. The discretization in time leads to the solution of the following sequence of LCPs:

$$\text{LCP}(\boldsymbol{B}^{(k+1)}, \boldsymbol{u}^{(k+1)}, \boldsymbol{b}^{(k+1)}, \boldsymbol{g}), \tag{8}$$

where $\boldsymbol{u}^{(k)}$ denotes the vector $\boldsymbol{u}$ at the kth time step. Here LCP($\boldsymbol{B}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{g}$) denotes the linear complementarity problem

$$\begin{cases} (\boldsymbol{B}\boldsymbol{u} - \boldsymbol{b}) \geq \boldsymbol{0}, \\ \boldsymbol{u} \geq \boldsymbol{g}, \\ (\boldsymbol{B}\boldsymbol{u} - \boldsymbol{b})^T (\boldsymbol{u} - \boldsymbol{g}) = 0. \end{cases}$$

For the first four time steps $k = 0, 1, 2, 3$, the implicit Euler method leads to

$$\boldsymbol{B}^{(k+1)} = \boldsymbol{I} + \frac{1}{2}\Delta\tau\boldsymbol{A} \qquad \text{and} \qquad \boldsymbol{b}^{(k+1)} = \boldsymbol{u}^{(k)} - \frac{1}{2}\Delta\tau\boldsymbol{a}. \tag{9}$$

For the rest of the time steps $k = 4, 5, \ldots, l-1$, the Crank-Nicolson method leads to

$$\boldsymbol{B}^{(k+1)} = \boldsymbol{I} + \frac{1}{2}\Delta\tau\boldsymbol{A} \qquad \text{and} \qquad \boldsymbol{b}^{(k+1)} = \left(\boldsymbol{I} - \frac{1}{2}\Delta\tau\boldsymbol{A}\right)\boldsymbol{u}^{(k)} - \Delta\tau\boldsymbol{a}. \tag{10}$$

4. The solution of LCPs

The projected SOR method (PSOR) for LCPs was introduced by Cryer in [28]. The method performs successive over relaxed corrections for the components of the solution vector combined with a projection when a component violates the early exercise constraint. For pricing American options this methods has been discussed in the books [29], [30], for example. The method is fairly simple to implement, but typically the number of iterations grows substantially when the discretization is refined. Thus, it is not usually efficient when fairly accurate option prices are sought. In this paper we will employ PSOR to (8) for comparison, see Sect. 5.

Here we will focus on an iterative scheme introduced in [22] and [25]. Let $\boldsymbol{B}$ denote the matrix $\boldsymbol{B}^{(k+1)}$ in (9) or (10) associated with the LCP (8). It has a regular splitting [31]

$$\boldsymbol{B} = \boldsymbol{T} - \boldsymbol{J},$$

where $\boldsymbol{J}$ is a block diagonal matrix with full diagonal blocks resulting from the integral term and $\boldsymbol{T}$ is the rest which is a block tridiagonal matrix. Based on this splitting, the first two authors of this paper proposed a fixed point iteration for LCPs in [22], [25]. It is a generalization of an iteration for linear systems described in [30] and applied in [15], [18], [20]. The fixed point iteration for LCP($\boldsymbol{B}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{g}$) reads

$$\text{LCP}(\boldsymbol{T}, \boldsymbol{u}^{j+1}, \boldsymbol{b} + \boldsymbol{J}\boldsymbol{u}^j, \boldsymbol{g}), \quad j = 0, 1, \ldots \tag{11}$$

Each iteration requires the solution of an LCP with the block tridiagonal $\boldsymbol{T}$ and the multiplication of a vector by $\boldsymbol{J}$. Below we will describe and compare PSOR and PAMG to solve these LCPs.

Based on a convergence result in [25] and the properties of the discretization, we can easily see that the reduction of the l_∞-norm of the error in each iteration of (11) is proportional to $\Delta\tau\lambda$. In general $\Delta\tau\lambda$ is much less than one yielding that the iteration converges very rapidly. In practice, typically only a few iterations are needed to reach sufficient accuracy for practical purposes.

One way to solve the LCPs in (11) is to use PSOR. Since it is not an efficient method for refined discretizations we will only use it here for comparison and instead make use of a projected algebraic multigrid method (PAMG) introduced in [26]. With a well designed multigrid method, the number of iterations does not grow with refined discretizations. For extensive literature on this see, the book [32], for example. For solving LCPs Brandt and Cryer introduced a projected full approximation scheme (PFAS) multigrid method in [33]. Another multigrid method for similar problems was described in [34]. The PFAS method was used to price American options under stochastic volatility by Clarke and Parrott in [9], and Oosterlee in [13]. Some alternative approaches employing multigrid methods for option pricing have been considered in [35], [36], [37]. Reisinger and Wittum described a projected multigrid (PMG) method for LCPs which resembles more closely a classical multigrid method for linear problems in [38]. This method has been used to price American options in [38], [11].

The above mentioned methods are so-called geometrical multigrid methods which means that the spatial operators are discretized on sequence of grids. Furthermore, transfer operators between grids need to be implemented.

The geometrical multigrid method can be implemented with some effort especially when the computational domain is a rectangle like in this case, but it is not a black-box method to which one can just give the matrix and vectors defining the LCP. An algebraic multigrid (AMG) method [39], [40] builds the coarse problems and the transfer operators automatically using the properties of the matrix. Recently, Toivanen and Oosterlee generalized an AMG method for LCPs and called the resulting method as the projected algebraic multigrid (PAMG) method [26]. Its treatment of LCPs in the coarser levels resemble the one in the PMG method [38]. The PAMG method is easy to use and efficient [26]. Below we present the algorithms for one iteration of PSOR and PAMG respectively.

Algorithm One iteration of PSOR($\boldsymbol{B}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{g}$)	**Algorithm** One iteration of PAMG($\boldsymbol{B}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{g}$)
for $i = 1, \ldots, \dim \boldsymbol{B}$	if coarsest level then
$\boldsymbol{r}_i = \boldsymbol{b}_i - \sum_{j=1}^{\dim \boldsymbol{B}} \boldsymbol{B}_{ij} \boldsymbol{u}_j$	solve LCP($\boldsymbol{B}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{g}$)
$\boldsymbol{u}_i = \boldsymbol{u}_i + \omega \boldsymbol{r}_i / \boldsymbol{B}_{ii}$	else
$\boldsymbol{u}_i = \max(\boldsymbol{u}_i, \boldsymbol{g}_i)$	PS($\boldsymbol{B}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{g}$)
end for	$\boldsymbol{u}^c = \boldsymbol{0}$
	$\boldsymbol{r}^c = \tilde{\boldsymbol{R}}(\boldsymbol{b} - \boldsymbol{B}\boldsymbol{u})$
	$\boldsymbol{g}^c = \hat{\boldsymbol{R}}(\boldsymbol{g} - \boldsymbol{u})$
	PAMG($\boldsymbol{B}^c, \boldsymbol{u}^c, \boldsymbol{r}^c, \boldsymbol{g}^c$)
	$\boldsymbol{u} = \boldsymbol{u} + \tilde{\boldsymbol{P}}\boldsymbol{u}^c$
	PS($\boldsymbol{B}, \boldsymbol{u}, \boldsymbol{b}, \boldsymbol{g}$)
	end if

Here $\tilde{\boldsymbol{R}}$ and $\hat{\boldsymbol{R}}$ denote the restriction operators for the solution of the LCP and its constraint respectively. The prolongation for the LCP is denoted by $\tilde{\boldsymbol{P}}$. Finally PS is a smoother for the LCP. For more details on these operators, see [26].

Finally, we summarize our algorithm to numerically price American options under the Bates model. Note that PSOR or PAMG form inner iterations to the outer LCP-iteration. In the next section we will see that for PAMG, both the outer and inner iteration-count is kept very low for each time-step.

Algorithm
Discretize (3) with (1) giving (8) with (9) and (10)
for $k = 1, \ldots, l$ (Time-stepping)
    for $j = 1, 2, \ldots$ until convergence (LCP-iteration)
        Solve (11) using PSOR or PAMG
    end for
end for

5. Numerical Experiment

In our numerical example, we price American call options. The parameters for the Bates model are the same as in [22] and they are defined below.

Parameter	Notation	Value
Risk free interest rate	r	0.03
Dividend yield	q	0.05
Strike price	K	100
Correlation between the price and variance processes	ρ	-0.5
Mean level of the variance	θ	0.04
Rate of reversion to the mean level	κ	2.0
Volatility of the variance	σ	0.25
Jump rate	λ	0.2
Mean jump	γ	-0.5
Variance of jump	δ	0.4

The computational domain is $(x, y, \tau) \in [0, 400] \times [0, 1] \times [0, 0.5]$. For the PSOR method, the relaxation parameter $\omega = 1.5$ is used.

In Table 1 we report the numerical results. The table has the following columns:

- Grid (m, n, l) defines the number of grid-points in x, y, and τ to be m, n, and l, respectively.
- LCP iter. gives the average number of LCP iterations on each time step.
- PSOR/PAMG iter. gives the average number of inner PSOR or PAMG iterations for solving one LCP.
- Error gives the root mean square relative error given by

$$\text{error} = \left[\frac{1}{5} \sum_{i=1}^{5} \left(\frac{u(\boldsymbol{x}_i, \theta, T) - U(\boldsymbol{x}_i, \theta, T)}{U(\boldsymbol{x}_i, \theta, T)} \right)^2 \right]^{1/2},$$

  where $\boldsymbol{x} = (80, 90, 100, 110, 120)^T$. The reference prices U given in [22] at $(\boldsymbol{x}_i, \theta, T)$, $i = 1, 2, \ldots, 5$ are 0.328526, 2.109397, 6.711622, 13.749337, 22.143307. They were computed using a componentwise splitting method on the grid (4096, 2048, 514).
- Ratio is the ratio of the consecutive root mean square relative errors.
- CPU gives the CPU time in seconds on a 2.0 GHz Intel Core i7 PC using one thread. For the PAMG method, the CPU time includes the AMG initialization time.

For the iterations, we use the stopping criterion

$$\left\| \bar{\boldsymbol{r}}^j \right\|_2 \leq 10^{-6} \|\boldsymbol{b}\|_2,$$

where $\bar{\boldsymbol{r}}^j$ is the reduced residual for the LCP iterations and the pure PSOR iterations respectively. It is defined by

$$\bar{\boldsymbol{r}}_i^j = \begin{cases} \boldsymbol{B}\boldsymbol{u}_i^j - \boldsymbol{b}_i & \text{if} \quad \boldsymbol{u}_i^j > \boldsymbol{g}_i \\ 0 & \text{if} \quad \boldsymbol{u}_i^j = \boldsymbol{g}_i. \end{cases}$$

For the inner PSOR/PAMG iterations it is defined similarly with $\boldsymbol{T}$ and the associated vectors instead of $\boldsymbol{B}$, $\boldsymbol{u}^j$, and $\boldsymbol{b}$.

The multiplication by the matrix $\boldsymbol{J}$ is the most expensive operation in the iteration. In order to perform it efficiently with the LCP iterations, we collected all $n + 1$ multiplications corresponding to all x-grid lines together and then performed the resulting matrix-matrix multiplication using the optimized GotoBLAS library [41].

In Table 1, roughly second-order accuracy is observed with the proposed discretization as the ratio is about four on average. On finer grids, only two LCP iterations are required to satisfy the stopping criterion. With the coarsest grid (64, 32, 10), the LCP iterations with the PSOR and PAMG methods require the same amount of time while the pure PSOR is twice slower. On finer grids, the speed differences become large and the number of PSOR iterations roughly doubles with each refinement. On the finest grid (1024, 512, 130), the LCP iteration with the PAMG method is about 12 times faster than the LCP iteration with the PSOR method, and it is 150 times faster than the pure PSOR iteration.

Table 1. The numerical results with five different space-time discretizations.

method	Grid (m, n, l)	LCP iter.	PSOR/PAMG iter.	error	ratio	CPU
PSOR	(64, 32, 10)		47.6	0.11765		0.09
	(128, 64, 18)		50.2	0.04068	2.89	1.24
	(256, 128, 34)		109.6	0.00740	5.49	42.00
	(512, 256, 66)		216.8	0.00227	3.26	1608.39
	(1024, 512, 130)		396.6	0.00038	5.93	251758.87
LCP iter.	(64, 32, 10)	2.4	35.5	0.11759		0.04
with PSOR	(128, 64, 18)	2.0	44.4	0.04064	2.89	0.28
	(256, 128, 34)	2.0	101.0	0.00738	5.50	5.19
	(512, 256, 66)	2.0	212.4	0.00226	3.27	107.91
	(1024, 512, 130)	2.0	417.0	0.00038	5.96	1673.90
LCP iter.	(64, 32, 10)	2.5	2.9	0.11760		0.04
with PAMG	(128, 64, 18)	2.1	3.5	0.04064	2.89	0.26
	(256, 128, 34)	2.0	3.5	0.00738	5.50	2.10
	(512, 256, 66)	2.0	3.3	0.00226	3.26	17.85
	(1024, 512, 130)	2.0	2.6	0.00038	5.93	133.11

6. Conclusions

In this paper we considered a linear complementarity problem (LCP) with a partial integro-differential operator for pricing American options under the Bates model. For the partial derivatives and integral we employed finite differences and simple quadrature respectively. In the numerical experiments, the discretizations are roughly second-order accurate in both space and time.

We proposed a rapidly converging iteration for solving LCPs at each time step. In each such iteration, an LCP with a sparse matrix needs to be solved. We demonstrated that these problems can be efficiently and easily solved with a projected algebraic multigrid method. With finer discretizations this approach leads to an order or several orders of magnitude faster method than using the projected SOR method.

References

[1] F. Black, M. Scholes, The pricing of options and corporate liabilities, J. Political Economy 81 (1973) 637–654.
[2] R. C. Merton, Option pricing when underlying stock returns are discontinuous, J. Financial Econ. 3 (1976) 125–144.
[3] S. G. Kou, A jump-diffusion model for option pricing, Management Sci. 48 (8) (2002) 1086–1101.
[4] S. Heston, A closed-form solution for options with stochastic volatility with applications to bond and currency options, Rev. Financial Stud. 6 (1993) 327–343.
[5] D. S. Bates, Jumps and stochastic volatility: Exchange rate processes implicit Deutsche mark options, Review Financial Stud. 9 (1) (1996) 69–107.
[6] D. Duffie, J. Pan, K. Singleton, Transform analysis and asset pricing for affine jump-diffusions, Econometrica 68 (6) (2000) 1343–1376.
[7] M. J. Brennan, E. S. Schwartz, The valuation of American put options, J. Finance 32 (1977) 449–462.
[8] S. Ikonen, J. Toivanen, Pricing American options using LU decomposition, Appl. Math. Sci. 1 (49-52) (2007) 2529–2551.
[9] N. Clarke, K. Parrott, Multigrid for American option pricing with stochastic volatility, Appl. Math. Finance 6 (1999) 177–195.
[10] S. Ikonen, J. Toivanen, Componentwise splitting methods for pricing American options under stochastic volatility, Int. J. Theor. Appl. Finance 10 (2) (2007) 331–361.
[11] S. Ikonen, J. Toivanen, Efficient numerical methods for pricing American options under stochastic volatility, Numer. Methods Partial Differential Equations 24 (1) (2008) 104–126.
[12] K. Ito, J. Toivanen, Lagrange multiplier approach with optimized finite difference stencils for pricing American options under stochastic volatility, SIAM J. Sci. Comput. 31 (4) (2009) 2646–2664.
[13] C. W. Oosterlee, On multigrid for linear complementarity problems with application to American-style options, Electron. Trans. Numer. Anal. 15 (2003) 165–185.
[14] R. Zvan, P. A. Forsyth, K. R. Vetzal, Penalty methods for American options with stochastic volatility, J. Comput. Appl. Math. 91 (2) (1998) 199–218.
[15] A. Almendral, C. W. Oosterlee, Numerical valuation of options with jumps in the underlying, Appl. Numer. Math. 53 (1) (2005) 1–18.
[16] L. Andersen, J. Andreasen, Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing, Rev. Deriv. Res. 4 (3) (2000) 231–262.
[17] R. Cont, E. Voltchkova, A finite difference scheme for option pricing in jump diffusion and exponential Lévy models, SIAM Numer. Anal. 43 (4) (2005) 1596–1626.
[18] Y. d'Halluin, P. A. Forsyth, K. R. Vetzal, Robust numerical methods for contingent claims under jump diffusion processes, IMA J. Numer. Anal. 25 (1) (2005) 87–112.

[19] A.-M. Matache, C. Schwab, T. P. Wihler, Fast numerical solution of parabolic integrodifferential equations with applications in finance, SIAM J. Sci. Comput. 27 (2) (2005) 369–393.
[20] J. Toivanen, Numerical valuation of European and American options under Kou's jump-diffusion model, SIAM J. Sci. Comput. 30 (4) (2008) 1949–1970.
[21] C. Chiarella, B. Kang, G. H. Meyer, A. Ziogas, The evaluation of American option prices under stochastic volatility and jump-diffusion dynamics using the method of lines, Int. J. Theor. Appl. Finance 12 (3) (2009) 393–425.
[22] J. Toivanen, A componentwise splitting method for pricing American options under the Bates model, in: Applied and numerical partial differential equations, Vol. 15 of Comput. Methods Appl. Sci., Springer, New York, 2010, pp. 213–227.
[23] L. Feng, V. Linetsky, Pricing options in jump-diffusion models: an extrapolation approach, Oper. Res. 56 (2) (2008) 304–325.
[24] R. Rannacher, Finite element solution of diffusion problems with irregular data, Numer. Math. 43 (2) (1984) 309–327.
[25] S. Salmi, J. Toivanen, An iterative method for pricing American options under jump-diffusion models, Appl. Numer. Math. 61 (7) (2011) 821–831.
[26] J. Toivanen, C. W. Oosterlee, A projected algebraic multigrid method for linear complementarity problems, Numer. Math. Theory Methods Appl. 5 (1) (2012) 85–98.
[27] M. B. Giles, R. Carter, Convergence analysis of Crank-Nicolson and Rannacher time-marching, J. Comput. Finance 9 (2006) 89–112.
[28] C. W. Cryer, The solution of a quadratic programming problem using systematic overrelaxation, SIAM J. Control 9 (1971) 385–392.
[29] R. U. Seydel, Tools for computational finance, 4th Edition, Universitext, Springer-Verlag, Berlin, 2009.
[30] D. Tavella, C. Randall, Pricing financial instruments: The finite difference method, John Wiley & Sons, Chichester, 2000.
[31] D. M. Young, Iterative solution of large linear systems, Academic Press, New York, 1971.
[32] U. Trottenberg, C. W. Oosterlee, A. Schüller, Multigrid, Academic Press Inc., San Diego, CA, 2001, with contributions by A. Brandt, P. Oswald and K. Stüben.
[33] A. Brandt, C. W. Cryer, Multigrid algorithms for the solution of linear complementarity problems arising from free boundary problems, SIAM J. Sci. Statist. Comput. 4 (4) (1983) 655–684.
[34] R. H. W. Hoppe, Multigrid algorithms for variational inequalities, SIAM J. Numer. Anal. 24 (5) (1987) 1046–1065.
[35] M. Holtz, A. Kunoth, B-spline-based monotone multigrid methods, SIAM J. Numer. Anal. 45 (3) (2007) 1175–1199.
[36] S. Ikonen, J. Toivanen, Operator splitting methods for pricing American options under stochastic volatility, Numer. Math. 113 (2) (2009) 299–324.
[37] A. Ramage, L. von Sydow, A multigrid preconditioner for an adaptive Black-Scholes solver, BIT 51 (1) (2011) 217–233.
[38] C. Reisinger, G. Wittum, On multigrid for anisotropic equations and variational inequalities: Pricing multi-dimensional European and American options, Comput. Vis. Sci. 7 (3-4) (2004) 189–197.
[39] J. W. Ruge, K. Stüben, Algebraic multigrid, in: Multigrid methods, Vol. 3 of Frontiers Appl. Math., SIAM, Philadelphia, PA, 1987, pp. 73–130.
[40] K. Stüben, Algebraic multigrid: An introduction with applications, in: Multigrid, Academic Press Inc., San Diego, CA, 2001.
[41] K. Goto, R. A. van de Geijn, Anatomy of high-performance matrix multiplication, ACM Trans. Math. Software 34 (3) (2008) Art. 12, 25.

PV

AN IMEX-SCHEME FOR PRICING OPTIONS UNDER STOCHASTIC VOLATILITY MODELS WITH JUMPS

by

S. Salmi, J. Toivanen and L. von Sydow 2013

Submitted for publication

An IMEX-scheme for pricing options under stochastic volatility models with jumps

Santtu Salmi* Jari Toivanen*† Lina von Sydow‡

November 5, 2013

Abstract

Partial integro-differential equation (PIDE) formulations are often preferable for pricing options under models with stochastic volatility and jumps, especially for American-style option contracts. We consider the pricing of options under such models, namely the Bates model and the so-called SVCJ model. The non-locality of the jump terms in these models lead to matrices with full matrix blocks. Standard discretization methods are not viable directly since they would require the inversion of such a matrix. Instead, we adopt a two-step implicit-explicit (IMEX) time discretization scheme, the IMEX-CNAB scheme, where the jump term is treated explicitly, while the rest is treated implicitly. The resulting linear systems can then be solved directly by employing LU decomposition. Alternatively, the systems can be iterated under a scalable algebraic multigrid (AMG) method. For pricing American options, LU decomposition is employed with an operator splitting for the early exercise constraint or, alternatively, a projected AMG method can be used to solve the resulting linear complementarity problems. We price European and American options in numerical experiments, which demonstrate the good efficiency of the proposed methods.

Keywords: option pricing, stochastic volatility model, jump-diffusion model, finite difference method, implicit-explicit time discretization

1 Introduction

Since the pioneering papers by Black and Scholes [6], and Merton [34], the shortcomings of their original model have become clear. It is well known that fitting empirically observed option prices into the Black-Scholes model typically implies a volatility distribution with a smile like shape. This volatility smile becomes more pronounced near the maturity date. The usual modifications of the Black-Scholes model to explain such implied volatility patterns include models with jumps and/or stochastic volatility.

While the underlying asset price can be completely modeled by an infinitude of predominantly small jumps, such as in the CGMY model [8], the conventional view is that jumps are relatively rare, high impact events. Over large time intervals the Brownian component

*Department of Mathematical Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland (santtu.salmi@jyu.fi, jari.toivanen@jyu.fi)

†Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305, USA (toivanen@stanford.edu)

‡Department of Information Technology, Uppsala University, Box 337, 751 05 Uppsala, Sweden (lina@it.uu.se)

becomes the dominant factor in the model since jumps are infrequent and their influence tends to cancel out. Thus, for options with long maturities the stochastic volatility models, for example the Heston model [23], are often regarded as more appropriate. For options with short maturities, however, jumps become increasingly important as a purely Brownian motion driven process would require extremely high levels of volatility to explain the pronounced volatility smile pattern. Well-known jump-diffusion models in the literature include the Merton [35] and Kou [29] models.

Bates proposed a more complete model in [5] with stochastic volatility and jumps in returns, essentially a combination of the Heston and Merton models. Duffie et al. [15] followed up with arguably a more realistic model with stochastic volatility and jumps in both returns and volatility. They considered independently arriving jumps (SVIJ) and contemporaneously arriving jumps (SVCJ). These models were further investigated by Eraker et al. in [16]. They concluded that SVIJ and SVCJ provide the best fit, but SVCJ is easier to estimate since jumps in returns and volatility are simultaneous.

In this paper we consider the numerical pricing of options under the Bates model (also known as the SVJ model) and the SVCJ model. A partial integro-differential equation (PIDE) can be derived for the price of a European option under the Bates and SVCJ models. Similarly, the price of an American option can be obtained by formulating a linear complementarity problem (LCP) with the same operator. Under the Heston model, numerical methods based on partial differential formulations for option pricing have been considered in the 1990s, for example, in [10, 53]. More recently similar methods have been generalized for the Bates model in [4, 9, 50, 36, 47, 40] and pricing European options under the SVJC model have been studied in [17, 52].

The finite difference and finite element methods are the most common ways to discretize the spatial operator; see [1]. For example, in [4, 17, 36, 40, 52] finite elements have been employd while in [9, 50, 47, 33, 38, 32, 39] finite differences are used. The discretization of the integral operator leads to matrices with full matrix blocks, for simplicity such matrices are called full matrices in the following. Standard implicit time discretization schemes lead to systems of equations with these full matrices. Efficient iterative methods can be used in combination with implicit discretization schemes, such as in [49, 44] for example. A more attractive approach is to employ special implicit-explicit (IMEX) time discretization schemes which treat the jump term explicitly and the rest implicitly. A first-order accurate IMEX-Euler scheme for option pricing under jump-diffusion models was proposed in [12]. Second-order accurate IMEX schemes, for example, in [3, 19] have been applied for option pricing in [17, 30, 31, 45]. In [46] a family of IMEX time discretization schemes was analyzed in the context of option pricing under jump-diffusion models. In particular the IMEX-CNAB scheme was found to be fast and accurate, while having desirable stability properties. Here we employ this time discretization scheme for the Bates and SVCJ models.

The implicit treatment of the partial differential operator leads to block tridiagonal systems which are of LCP form for American options. While the projected SOR method (PSOR) [13] can be easily employed to solve these systems, its efficiency deteriorates, i.e. the number of required iterations grows when discretizations are refined. A scalable alternative is a well designed multigrid method which require a constant number of iterations also with refined discretizations. The first such method for LCPs was the projected full approximation scheme (PFAS) multigrid method introduced by Brandt and Cryer in [7]. This method was used to price American options by Clarke and Parrott in [10], and Oosterlee in [37]. The projected multigrid (PMG) method for LCPs introduced by Reisinger and

Wittum in [42] resembles more closely multigrid methods for linear systems. The PFAS and PMG methods are so-called geometrical multigrid methods which require a sequence of hierarchical grids. An easier-to-use alternative is algebraic multigrid (AMG) methods [43, 48] which automatically generate a sequence of coarser problems. The projected algebraic multigrid (PAMG) method proposed in [51] is a generalization of these algebraic methods for LCPs. This is one of the methods that we employ for the solution of LCPs.

The other method that we will use is the operator splitting method proposed in [24] and employed for the Heston model in [26, 27]. This method approximates the LCP as a system of linear equations and a set of simple decoupled linear complementarity problems. A popular alternative approximation is the penalty method which was considered for the Heston model by Zvan, Forsyth, and Vetzal in [53] and since then by many authors for various option pricing models. For the block tridiagonal systems of linear equations many different methods can employed. In this paper, we will show that these systems can be solved very efficiently using a modern sparse direct solver when the time-step is constant. In this case, an LU decomposition of the coefficient-matrix needs to be formed only once. For two-dimensional problems like the underlying one, George showed in his 1973 paper [21] that the decomposition can be formed using $\mathcal{O}(m^3)$ operations and the solution each time-step requires $\mathcal{O}(m^2 \log m)$ operations, where m is of the same order as the number of grid-points in both directions. As the number of time-steps is usually also of the same order, i.e. $\sim m$, the computational complexity of the time-stepping is greater than the one required by forming the LU decomposition.

Explicit treatment of the jumps leads to matrix-vector multiplications with a full matrix. These multiplications can be performed efficiently by employing FFT based implementations. For a logarithmically uniform grid, the jump matrix under the SVJ model is a Toeplitz matrix, which can be embedded into a circulant matrix, as in [2] for example. The multiplication can then be computed by an FFT and an inverse FFT, which require only $\mathcal{O}(m \log m)$ operations for each grid line. This approach is more involved for the SVCJ model, where FFTs need to be performed in two directions. Again under the SVCJ model the jump matrix can be embedded into a block circulant matrix with circulant blocks (BCCB); see [17], for example. The FFTs in both directions require $\mathcal{O}(m^2 \log m)$ operations. Here we describe and employ this approach with the SVCJ model.

2 Option pricing model

2.1 Governing equations

First we consider a model with stochastic volatility and jumps in returns described by Bates [5]. Under this model the behavior of the asset value s and its variance v is described by the coupled stochastic differential equations

$$\begin{aligned} ds &= \mu_B s dt + \sqrt{v} s dw_1 + s dJ, \\ dv &= \kappa(\theta - v)dt + \sigma\sqrt{v} dw_2. \end{aligned} \tag{1}$$

Here μ_B is the drift rate defined by $\mu_B = r - q - \lambda\xi_B$ where $r \geq 0$ is the risk-free interest rate and $q \geq 0$ is the continuous dividend yield. The jump process J is a compound Poisson process with intensity $\lambda > 0$ and $J + 1$ has a log-normal distribution $p(y)$ with the mean in $\log y$ being γ and the variance in $\log y$ being δ^2, i.e. the probability density function is given by $p(y) = \frac{1}{\sqrt{2\pi} y \delta} e^{-\frac{(\log y - \gamma)^2}{2\delta^2}}$. The parameter ξ_B is defined by $\xi_B = e^{\gamma + \delta^2/2} - 1$. The

variance v has mean level θ, κ is the rate of reversion to the mean level of v, and σ is the volatility of the variance v. The two Wiener processes w_1 are w_2 have the correlation ρ.

By combining derivative pricing arguments from [11, 18] for the Bates model, we can obtain the PIDE (formulated in forward time)

$$\begin{aligned}\frac{\partial u}{\partial \tau} &= \frac{1}{2}vs^2\frac{\partial^2 u}{\partial s^2} + \rho\sigma vs\frac{\partial^2 u}{\partial s \partial v} + \frac{1}{2}\sigma^2 v\frac{\partial^2 u}{\partial v^2} + (r - q - \lambda\xi_B)s\frac{\partial u}{\partial s} + \kappa(\theta - v)\frac{\partial u}{\partial v} \\ &\quad - (r+\lambda)u + \lambda\int_0^\infty u(sy, v, \tau)p(y)dy =: L_D^B u + L_I^B u,\end{aligned} \tag{2}$$

where u is the price of a European option and $\tau = T - t$ is the time to expiry. The operators L_D^B and L_I^B are defined as the differential part (including the term $-(r+\lambda)u$) and the integral part of (2), respectively. The initial condition for (2) is defined by

$$u = g(s, v), \tag{3}$$

where g is the pay-off function which gives the value of the option at the expiry.

Next, we allow for jumps in the volatility and study SVCJ. Then we have

$$\begin{aligned}ds &= \mu_S s dt + \sqrt{v}s dw_1 + s dJ^s \\ dv &= \kappa(\theta - v)dt + \sigma\sqrt{v}dw_2 + dJ^v.\end{aligned} \tag{4}$$

Now $\mu_S = r - q - \lambda\xi_S$ where ξ_S is defined by $\xi_S = e^{\gamma+\delta^2/2}(1-\nu\rho_J)^{-1} - 1$ and ρ_J defines the correlation between jumps in returns and variance. The two-dimensional jump process (J^s, J^v) is a $\mathbb{R}\times\mathbb{R}^+$-valued compound Poisson process with intensity $\lambda > 0$. The distribution of the jump size in variance is assumed to be exponential with mean ν. Conditional on a jump of size z^v in the variance process, J^s+1 has a log-normal distribution $p(z^s, z^v)$ with the mean in $\log z^s$ being $\gamma + \rho_J z^v$. This gives a bivariate probability density function defined by $p(z^s, z^v) = \frac{1}{\sqrt{2\pi}z^s\delta\nu}e^{-\frac{z^v}{\nu} - \frac{(\log z^s - \gamma - \rho_J z^v)^2}{2\delta^2}}$.

As in [15, 17], we assume that the price of a European option under the SVCJ-model can be obtained as the solution to the PIDE

$$\begin{aligned}\frac{\partial u}{\partial \tau} &= \frac{1}{2}vs^2\frac{\partial^2 u}{\partial s^2} + \rho\sigma vs\frac{\partial^2 u}{\partial s \partial v} + \frac{1}{2}\sigma^2 v\frac{\partial^2 u}{\partial v^2} + (r - q - \lambda\xi_S)s\frac{\partial u}{\partial s} + \kappa(\theta - v)\frac{\partial u}{\partial v} \\ &\quad - (r+\lambda)u + \lambda\int_0^\infty\int_0^\infty u(s\cdot z^s, v + z^v, \tau)p(z^s, z^v)dz^v dz^s =: L_D^S u + L_I^S u.\end{aligned} \tag{5}$$

The initial condition is defined by (3).

For computational reasons we truncate the unbounded domain to $(s, v, \tau) \in (0, s_{\max})\times(0, v_{\max})\times(0, T]$. We impose the boundary conditions $u(0, v, \tau) = e^{-r\tau}g(0, v)$, $u(s_{\max}, v, \tau) = g(s_{\max}, v)$, and $\frac{\partial u}{\partial v}(s, v_{\max}, \tau) = 0$. On the boundary $v = 0$, the PIDEs (2) and (5) hold and no additional boundary condition needs to be posed; see [27] on discussion on this boundary and its treatment.

2.2 Linear complementarity problem for American options

In the following, L_D and L_I denote the differential operator L_D^B or L_D^S, and the integral operator L_I^B or L_I^S, respectively. With this notation both (2) and (5) can be expressed as

$\frac{\partial u}{\partial \tau} - L_D u - L_I u =: \mathcal{L}u = 0$. The price u of an American option satisfies the LCP defined by

$$\begin{cases} \mathcal{L}u \geq 0, \quad u \geq 0, \\ (\mathcal{L}u)(u-g) = 0. \end{cases} \tag{6}$$

We impose the same boundary conditions as for (2) and (5) except on the boundary $s = 0$ the boundary condition is without discounting, i.e. $u(0, v, \tau) = g(0, v)$.

3 Discretization

3.1 Computational grid

We will use a computational grid that is uniform in τ and nonuniform in s and v. The spatial grid is denoted (s_i, v_j), $i = 1, \ldots, m_s$, $j = 1, \ldots, m_v$. We employ grid generating functions $s : [0,1] \to [0, s_{\max}]$ and $v : [0,1] \to [0, v_{\max}]$ to define the grid-points as $s_i = s((i-1)/(m_s - 1))$ and $v_j = v((j-1)/(m_v - 1))$.

There are many ways to choose the functions s and v. Here we use the quadratic functions

$$s(p) = ap^2 + bp, \;\; p \in [0,1], \qquad v(q) = cq^2 + dq, \;\; q \in [0,1] \tag{7}$$

with the coefficients a, b, c, and d defined by the following conditions. In order to have the end points at $s_{\max}$ and $v_{\max}$ the conditions $s(1) = s_{\max}$ and $v(1) = v_{\max}$ have to hold. We choose s to map the point p_K to the strike price K, i.e. $s(p_K) = K$. When this point satisfies $p_K < K/s_{\max}$ the grid is refined in the interval $[0, K]$. A similar condition can be defined also for v, but instead we require the grid to be α times finer at 0 than at $v_{\max}$. This leads to the condition $v'(0) = \alpha v'(1)$. These four conditions define the coefficient a, b, c, and d.

At a grid point $s_i = s((i-1)/(m_s - 1))$ the grid step to the left is denoted by $\Delta s_i^- = s_i - s_{i-1}$ and the grid step to the right is denoted by $\Delta s_i^+ = s_{i+1} - s_i$. The grid steps around v_j are denoted in the same way.

3.2 IMEX-discretization in time

In [46] a number of IMEX-discretization methods for option pricing problems under jump-diffusion models are examined. The authors found that the IMEX-CNAB method produced the smallest error among the methods they studied. This method can be seen as a modification of the popular Crank-Nicolson method with a second-order accurate explicit threatment for the integral operator. For these reasons, we will employ and promote this method here.

We start by taking a small even number $2\tilde{N}$ of Rannacher-style smoothing Euler steps [41, 22] with the time-step $\Delta\tau/2$ given by

$$\left(I - \frac{\Delta\tau}{2} L_D\right) u^{(n+1)/2} = \left(I + \frac{\Delta\tau}{2} L_I\right) u^{n/2}, \quad n = 0, \ldots, 2\tilde{N} - 1. \tag{8}$$

This is followed by IMEX-CNAB steps defined by

$$\left(I - \frac{\Delta\tau}{2} L_D\right) u^{n+1} = \left(I + \frac{\Delta\tau}{2} L_D + \frac{3\Delta\tau}{2} L_I\right) u^n - \frac{\Delta\tau}{2} L_I u^{n-1}, \quad n = \tilde{N}, \ldots, N. \tag{9}$$

3.3 Discretization of the differential operator

Here we construct a seven-point finite difference discretization for the differential operator

$$\begin{aligned} L_D u = &\frac{1}{2} v s^2 \frac{\partial^2 u}{\partial s^2} + \rho\sigma v s \frac{\partial^2 u}{\partial s \partial v} + \frac{1}{2}\sigma^2 v \frac{\partial^2 u}{\partial v^2} \\ &+ (r - q - \lambda\xi) s \frac{\partial u}{\partial s} + \kappa(\theta - v)\frac{\partial u}{\partial v} - (r+\lambda)u. \end{aligned} \tag{10}$$

We start with the mixed derivative and assume that the correlation ρ is negative. A similar seven-point finite difference discretization can also be constructed for a positive ρ; see [25, 26], for example. An alternative approach to construct finite difference stencils is considered in [28]. A Taylor-expansion of $u(s_{i+1}, v_{j-1}, \tau)$ and $u(s_{i-1}, v_{j+1}, \tau)$ around (s_i, v_j, τ) leads to, omitting the argument (s_i, v_j, τ) in the expressions

$$\begin{aligned} u(s_{i+1}, v_{j-1}, \tau) \approx\ & u + \Delta s_i^+ \frac{\partial u}{\partial s} - \Delta v_j^- \frac{\partial u}{\partial v} \\ &+ \frac{(\Delta s_i^+)^2}{2}\frac{\partial^2 u}{\partial s^2} - \Delta s_i^+ \Delta v_j^- \frac{\partial^2 u}{\partial s \partial v} + \frac{(\Delta v_j^-)^2}{2}\frac{\partial^2 u}{\partial v^2}, \\ u(s_{i-1}, v_{j+1}, \tau) \approx\ & u - \Delta s_i^- \frac{\partial u}{\partial s} + \Delta v_j^+ \frac{\partial u}{\partial v} \\ &+ \frac{(\Delta s_i^-)^2}{2}\frac{\partial^2 u}{\partial s^2} - \Delta s_i^- \Delta v_j^+ \frac{\partial^2 u}{\partial s \partial v} + \frac{(\Delta v_j^+)^2}{2}\frac{\partial^2 u}{\partial v^2}, \end{aligned} \tag{11}$$

which gives

$$\begin{aligned} \frac{\partial^2 u}{\partial s \partial v} &\approx \frac{-u(s_{i+1}, v_{j-1}, \tau) + u + \Delta s_i^+ \frac{\partial u}{\partial s} - \Delta v_j^- \frac{\partial u}{\partial v} + \frac{(\Delta s_i^+)^2}{2}\frac{\partial^2 u}{\partial s^2} + \frac{(\Delta v_j^-)^2}{2}\frac{\partial^2 u}{\partial v^2}}{\Delta s_i^+ \Delta v_j^-}, \\ \frac{\partial^2 u}{\partial s \partial v} &\approx \frac{-u(s_{i-1}, v_{j+1}, \tau) + u - \Delta s_i^- \frac{\partial u}{\partial s} + \Delta v_j^+ \frac{\partial u}{\partial v} + \frac{(\Delta s_i^-)^2}{2}\frac{\partial^2 u}{\partial s^2} + \frac{(\Delta v_j^+)^2}{2}\frac{\partial^2 u}{\partial v^2}}{\Delta s_i^- \Delta v_j^+}. \end{aligned} \tag{12}$$

Using a weighted average of the approximations in (12) we obtain

$$\begin{aligned} \frac{\partial^2 u}{\partial s \partial v} \approx\ & w \frac{-u(s_{i+1}, v_{j-1}, \tau) + u + \Delta s_i^+ \frac{\partial u}{\partial s} - \Delta v_j^- \frac{\partial u}{\partial v} + \frac{(\Delta s_i^+)^2}{2}\frac{\partial^2 u}{\partial s^2} + \frac{(\Delta v_j^-)^2}{2}\frac{\partial^2 u}{\partial v^2}}{\Delta s_i^+ \Delta v_j^-} \\ &+ (1-w)\frac{-u(s_{i-1}, v_{j+1}, \tau) + u - \Delta s_i^- \frac{\partial u}{\partial s} + \Delta v_j^+ \frac{\partial u}{\partial v} + \frac{(\Delta s_i^-)^2}{2}\frac{\partial^2 u}{\partial s^2} + \frac{(\Delta v_j^+)^2}{2}\frac{\partial^2 u}{\partial v^2}}{\Delta s_i^- \Delta v_j^+}. \end{aligned} \tag{13}$$

Using (13) in (10), we obtain the approximation

$$\begin{aligned} L_D u \approx\ & c_{ss}\frac{\partial^2 u}{\partial s^2} + c_{vv}\frac{\partial^2 u}{\partial v^2} + c_s \frac{\partial u}{\partial s} + c_v \frac{\partial u}{\partial v} \\ &+ \left(-(r+\lambda) + \rho\sigma v s \left(\frac{w}{\Delta s_i^+ \Delta v_j^-} + \frac{1-w}{\Delta s_i^- \Delta v_j^+} \right) \right) u \\ &- \frac{w}{\Delta s_i^+ \Delta v_j^-} u(s_{i+1}, v_{j-1}, \tau) - \frac{1-w}{\Delta s_i^- \Delta v_j^+} u(s_{i-1}, v_{j+1}, \tau), \end{aligned} \tag{14}$$

where the coefficients are

$$
\begin{aligned}
c_{ss} &= \frac{1}{2}\left(vs^2 + \rho\sigma vs\left(\frac{w\Delta s_i^+}{\Delta v_j^-} + \frac{(1-w)\Delta s_i^-}{\Delta v_j^+}\right)\right),\\
c_{vv} &= \frac{1}{2}\left(\sigma^2 v + \rho\sigma vs\left(\frac{w\Delta v_j^-}{\Delta s_i^+} + \frac{(1-w)\Delta v_j^+}{\Delta s_i^-}\right)\right),\\
c_s &= (r-q-\lambda\xi)s + \rho\sigma vs\left(\frac{w}{\Delta v_j^-} - \frac{1-w}{\Delta v_j^+}\right),\\
c_v &= \kappa(\theta - v) + \rho\sigma vs\left(-\frac{w}{\Delta s_i^+} + \frac{1-w}{\Delta s_i^-}\right).
\end{aligned}
\tag{15}
$$

Second-order finite difference discretizations of the second-derivatives are defined by

$$
\begin{aligned}
\frac{\partial^2 u}{\partial s^2}(s_i, v_j, \tau) &\approx 2\frac{\frac{1}{\Delta s_i^+}u(s_{i+1}, v_j, \tau) - \left(\frac{1}{\Delta s_i^+} + \frac{1}{\Delta s_i^-}\right)u(s_i, v_j, \tau) + \frac{1}{\Delta s_i^-}u(s_{i-1}, v_j, \tau)}{\Delta s_i^- + \Delta s_i^+},\\
\frac{\partial^2 u}{\partial v^2}(s_i, v_j, \tau) &\approx 2\frac{\frac{1}{\Delta v_j^+}u(s_i, v_{j+1}, \tau) - \left(\frac{1}{\Delta v_j^+} + \frac{1}{\Delta v_j^-}\right)u(s_i, v_j, \tau) + \frac{1}{\Delta v_j^-}u(s_i, v_{j-1}, \tau)}{\Delta v_j^- + \Delta v_j^+}.
\end{aligned}
\tag{16}
$$

Similarly, second-order finite differences for the first-order derivatives are defined by

$$
\begin{aligned}
\frac{\partial u}{\partial s}(s_i, v_j, \tau) &\approx \frac{\frac{\Delta s_i^-}{\Delta s_i^+}u(s_{i+1}, v_j, \tau) - \left(\frac{\Delta s_i^-}{\Delta s_i^+} - \frac{\Delta s_i^+}{\Delta s_i^-}\right)u(s_i, v_j, \tau) - \frac{\Delta s_i^+}{\Delta s_i^-}u(s_{i-1}, v_j, \tau)}{\Delta s_i^- + \Delta s_i^+},\\
\frac{\partial u}{\partial v}(s_i, v_j, \tau) &\approx \frac{\frac{\Delta v_j^-}{\Delta v_j^+}u(s_i, v_{j+1}, \tau) - \left(\frac{\Delta v_j^-}{\Delta v_j^+} - \frac{\Delta v_j^+}{\Delta v_j^-}\right)u(s_i, v_j, \tau) - \frac{\Delta v_j^+}{\Delta v_j^-}u(s_i, v_{j-1}, \tau)}{\Delta v_j^- + \Delta v_j^+}.
\end{aligned}
\tag{17}
$$

When a finite difference method is employed, spurious oscillations might occur when the discretization matrix of $-L_D$ does not lead to a so called M-matrix. Sufficient conditions for an M-matrix are that it is strictly diagonally dominant with positive diagonal elements and it has non-positive off-diagonal elements. To ensure the off-diagonals to be non-positive, we add artificial diffusion when necessary. With the second-order differences (16) and (17), this leads to the modified diffusion coefficients

$$
\begin{aligned}
\tilde{c}_{ss} &= \max\left\{\tilde{c}_{ss},\ -\frac{1}{2}c_s\Delta s_i^-,\ \frac{1}{2}c_s\Delta s_i^+\right\},\\
\tilde{c}_{vv} &= \max\left\{\tilde{c}_{vv},\ -\frac{1}{2}c_v\Delta v_i^-,\ \frac{1}{2}c_v\Delta v_i^+\right\}.
\end{aligned}
\tag{18}
$$

When the original coefficients c_{ss} and c_{vv} are positive, this addition of artificial diffusion is equivalent to using one-sided first-order finite differences to discretize a part of the first-order derivatives.

In order not to add excessive diffusion to the discretization, it is desirable that the coefficients c_{ss} and c_{vv} are non-negative. For a general weight $w \in [0,1]$, this leads to the bounds

$$
-\frac{\rho}{\sigma}s_i\Delta v_j^+ \le \Delta s_i^- \le -\frac{1}{\rho\sigma}s_i\Delta v_j^+ \quad \text{and} \quad -\frac{\rho}{\sigma}s_i\Delta v_j^- \le \Delta s_i^+ \le -\frac{1}{\rho\sigma}s_i\Delta v_j^- \tag{19}
$$

for the grid step sizes Δs_i^- and Δs_i^+. These bounds can be quite stringent when the correlation ρ approaches -1.

Note that the diagonal element of the discretization matrix is $r + \lambda$ minus the sum of the off-diagonal elements. Thus, the matrix is strictly diagonally dominant with positive diagonals when $r + \lambda > 0$ and the off-diagonal elements are non-positive.

3.4 Discretization of the integral operator

3.4.1 Bates' model

The integral term can be written as

$$I = \int_0^\infty u(sy, v, \tau)p(y)dy = \int_{-\infty}^\infty \bar{u}(z + x, v, \tau)\bar{p}(z)dz, \tag{20}$$

where $x = \log s$, $z = \log y$, $\bar{u}(z, v, \tau) = u(e^z, v, \tau)$, and $\bar{p}(z) = p(e^z)e^z$. Now, make the change of variable $\zeta = z + x$ and study the value of the integral at the point x_i

$$\begin{aligned} I_i = \int_{-\infty}^\infty \bar{u}(\zeta, v, \tau)\bar{p}(\zeta - x_i)d\zeta = \int_{x_{\min}}^{x_{\max}} \bar{u}(\zeta, v, \tau)\bar{p}(\zeta - x_i)d\zeta \\ + \int_{-\infty}^{x_{\min}} \bar{g}(\zeta, v, \tau)\bar{p}(\zeta - x_i)d\zeta + \int_{x_{\max}}^\infty \bar{g}(\zeta, v, \tau)\bar{p}(\zeta - x_i)d\zeta. \end{aligned} \tag{21}$$

The interval $(x_{\min}, x_{\max})$ is chosen to be so large that the two last integrals of (21) are negligible. The first part of the integral is evaluated using the trapezoidal quadrature rule on an equidistant grid in x with spacing Δx and m_x grid-points in $(x_{\min}, x_{\max})$ giving

$$I_i \approx \int_{x_{\min}}^{x_{\max}} \bar{u}(\zeta, v, \tau)\bar{p}(\zeta - x_i)d\zeta \approx \Delta x \sum_{j=1}^{m_x} \bar{u}(\zeta_j, v, \tau)\bar{p}(\zeta_j - x_i) = \bar{I}_i. \tag{22}$$

Note that the above appoximation includes the additional terms $\frac{\Delta x}{2}\bar{u}(x_{\min}, v, \tau)\bar{p}(x_{\min} - x_i)$ and $\frac{\Delta x}{2}\bar{u}(x_{\max}, v, \tau)\bar{p}(x_{\max} - x_i)$ which are also assumed to be negligible. Computing all $\bar{I}_i$, $i = 1, \ldots, m_x$ can be accomplished by the matrix-vector multiplication defined by

$$\bar{\mathbf{I}} = \mathbf{T}_{m_x}\bar{\mathbf{u}},$$

where

$$\bar{\mathbf{I}} = \begin{pmatrix} \bar{I}_1 & \bar{I}_2 & \cdots & \bar{I}_{m_x-1} & \bar{I}_{m_x} \end{pmatrix}^T, \quad \bar{\mathbf{u}} = \begin{pmatrix} \bar{u}_1 & \bar{u}_2 & \cdots & \bar{u}_{m_x-1} & \bar{u}_{m_x} \end{pmatrix}^T,$$

$$\mathbf{T}_{m_x} = \begin{pmatrix} \bar{p}(0) & \bar{p}(\Delta x) & \cdots & \bar{p}((m_x-2)\Delta x) & \bar{p}((m_x-1)\Delta x) \\ \bar{p}(-\Delta x) & \bar{p}(0) & \bar{p}(\Delta x) & \cdots & \bar{p}((m_x-2)\Delta x) \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \bar{p}(-(m_x-2)\Delta x) & \cdots & \bar{p}(-\Delta x) & \bar{p}(0) & \bar{p}(\Delta x) \\ \bar{p}(-(m_x-1)\Delta x) & \bar{p}(-(m_x-2)\Delta x) & \cdots & \bar{p}(-\Delta x) & \bar{p}(0) \end{pmatrix}.$$

The matrix $\mathbf{T}_{m_x}$ is a so called Toeplitz matrix with constant diagonals. Such matrices can be embedded in circulant matrices which for $\mathbf{T}_{m_x}$ yields

$$\mathbf{C}_{2m_x-1} = \left(\begin{array}{ccc|c|ccc} \bar{p}(0) & \cdots & \bar{p}((m_x-1)\Delta x) & \cdots & \bar{p}(-\Delta x) \\ \vdots & \ddots & & \ddots & \vdots \\ \bar{p}(-(m_x-1)\Delta x) & & \bar{p}(0) & & \bar{p}((m_x-1)\Delta x) \\ \hline \vdots & \ddots & & \ddots & \vdots \\ \bar{p}(\Delta x) & \cdots & \bar{p}(-(m_x-1)\Delta x) & \cdots & \bar{p}(0) \end{array}\right).$$

The computation of $\bar{\mathbf{I}} = \mathbf{T}_{m_x}\bar{\mathbf{u}}$ can be performed by first computing $\tilde{\mathbf{I}} = \mathbf{C}_{2m_x-1}\tilde{\mathbf{u}}$, where $\tilde{\mathbf{u}} = \begin{pmatrix}\bar{u}_1 & \bar{u}_2 & \cdots & \bar{u}_{m_x-1} & \bar{u}_{m_x} & 0 & \cdots & 0\end{pmatrix}^T$. Then $\bar{\mathbf{I}}$ is given by the m_x first elements in $\tilde{\mathbf{I}}$.

The circulant matrix $\mathbf{C}_{2m_x-1}$ can be decomposed as $\mathbf{C}_{2m_x-1} = \mathbf{F}^{-1}_{2m_x-1}\mathbf{\Lambda}\mathbf{F}_{2m_x-1}$, where $\mathbf{F}_{2m_x-1}$ is a Fourier matrix of order $2m_x-1$ and $\mathbf{\Lambda}$ is a diagonal matrix with the eigenvalues of $\mathbf{C}_{2m_x-1}$ on the diagonal. These eigenvalues can be computed by $\mathbf{F}_{2m_x-1}\mathbf{c}$ where $\mathbf{c}$ is the first column vector in $\mathbf{C}_{2m_x-1}$. From this we conclude that the multiplication of a vector $\mathbf{w}$ by the matrix $\mathbf{C}_{2m_x-1}$ can be computed as $\mathbf{F}^{-1}_{2m_x-1}\mathbf{\Lambda}\mathbf{F}_{2m_x-1}\mathbf{w}$. This can then be accomplished using two discrete Fourier transforms (DFTs) and one inverse discrete Fourier transform (IDFT). By embedding $\mathbf{T}_{m_x}$ in a circulant matrix $\mathbf{C}_{M_x}$ where M_x is the smallest power of 2 such that $M_x \geq 2m_x - 1$, the DFTs and IDFTs can be computed using fast Fourier transforms (FFTs) requiring $\mathcal{O}(M_x \log_2 M_x)$ arithmetic operations. If there are repeated matrix-vector multiplications with the matrix $\mathbf{C}_{M_x}$, then the eigenvalues of $\mathbf{C}_{M_x}$ can be precomputed once in the beginning.

We summarize the computation of the integral in (20) as follows:

- Interpolate values from the s_i grid-points to equidistant points x_i between $x_{\min}$ and $x_{\max}$.
- Compute and embed the matrix $\mathbf{T}_{m_x}$ into a circulant matrix $\mathbf{C}_{M_x}$.
- Compute $\bar{\mathbf{I}}$ using the algorithm described above using FFTs.
- Interpolate $\bar{I}_i$ to the original grid-points s_i.

3.4.2 SVCJ model

The integral term can be written as

$$I = \int_0^\infty\int_0^\infty u(sz^s, v+z^v, \tau)p(z^s, z^v)dz^v dz^s = \int_{-\infty}^\infty\int_0^\infty \bar{u}(x+z^x, v+z^v, \tau)\bar{p}(z^x, z^v)dz^v dz^x, \quad (23)$$

where $x = \log s$, $z^x = \log z^s$, $\bar{u}(x, v, \tau) = u(e^x, v, \tau)$, and $\bar{p}(z^x, z^v) = p(e^{z^x}, z^v)e^{z^x}$.

Now, make the changes of variables $\zeta = x + z^x$ and $\eta = v + z^v$ and study the value of the integral at the point (x_i, v_j)

$$I_{i,j} = \int_{-\infty}^\infty\int_{v_j}^\infty \bar{u}(\zeta, \eta, \tau)\bar{p}(\zeta - x_i, \eta - v_j)d\eta d\zeta. \quad (24)$$

Similarly to the treatment of the integral in the Bates model, we choose a rectangle $(x_{\min}, x_{\max}) \times (v_j, \bar{v}_{\max})$ which is large enough so that integrating over it gives a sufficiently good approximation for (24). The first part of the integral is evaluated using the two-dimensional generalization of the trapezoidal rule on an equidistant grid in x and in v with spacing Δv and $\bar{m}_v$ grid-points giving

$$\begin{aligned}&\int_{x_{\min}}^{x_{\max}}\int_{v_j}^{\bar{v}_{\max}} \bar{u}(\zeta, \eta, \tau)\bar{p}(\zeta - x_i, \eta - v_j)d\eta d\zeta \approx \frac{\Delta x \Delta v}{2}\sum_{k=1}^{m_x} \bar{u}(\zeta_k, \eta_j, \tau)\bar{p}(\zeta_k - x_i, 0)\\ &+ \Delta x \Delta v \sum_{k=1}^{m_x}\sum_{\ell=j+1}^{\bar{m}_v} \bar{u}(\zeta_k, \eta_\ell, \tau)\bar{p}(\zeta_k - x_i, \eta_\ell - v_j) = \bar{I}_{i,j}.\end{aligned} \quad (25)$$

By defining $\bar{\mathbf{I}} = \left(\bar{\mathbf{I}}_1\, \bar{\mathbf{I}}_2 \cdots \bar{\mathbf{I}}_{m_x-1}\, \bar{\mathbf{I}}_{m_x}\right)^T$, $\bar{\mathbf{u}} = \left(\bar{\mathbf{u}}_1\, \bar{\mathbf{u}}_2 \cdots \bar{\mathbf{u}}_{m_x-1}\, \bar{\mathbf{u}}_{m_x}\right)^T$, $\bar{\mathbf{I}}_\ell = \left(\bar{I}_{1,\ell}\, \bar{I}_{2,\ell} \cdots \bar{I}_{\bar{m}_v-1,\ell}\, \bar{I}_{\bar{m}_v,\ell}\right)^T$, $\bar{\mathbf{u}}_\ell = \left(\bar{u}_{1,\ell}\, \bar{u}_{2,\ell} \cdots \bar{u}_{\bar{m}_v-1,\ell}\, \bar{u}_{\bar{m}_v,\ell}\right)^T$, we can compute $\bar{\mathbf{I}}$ by $\bar{\mathbf{I}} = \mathbf{T}_{\bar{m}_v m_x}\bar{\mathbf{u}}$, where $\mathbf{T}_{\bar{m}_v m_x}$ is a block-Toeplitz matrix with Toeplitz blocks (BTTB-matrix) defined by

$$\mathbf{T}_{\bar{m}_v m_x} = \begin{pmatrix} \mathbf{T}_0 & \mathbf{T}_1 & \cdots & \mathbf{T}_{\bar{m}_v-1} \\ \mathbf{T}_{-1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{T}_1 \\ \mathbf{T}_{-(\bar{m}_v-1)} & \cdots & \mathbf{T}_{-1} & \mathbf{T}_0 \end{pmatrix},\ \mathbf{T}_\ell = \begin{pmatrix} T_{0,\ell} & T_{1,\ell} & \cdots & T_{m_x-1,\ell} \\ T_{-1,\ell} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & T_{1,\ell} \\ T_{-(m_x-1),\ell} & \cdots & T_{-1,\ell} & T_{0,\ell} \end{pmatrix}, \quad (26)$$

where

$$T_{k,\ell} = \begin{cases} 0, & \ell < 0, \\ \frac{1}{2}\Delta x \Delta v \bar{p}(k\Delta x, 0), & \ell = 0, \\ \Delta x \Delta v \bar{p}(k\Delta x, \ell\Delta v), & \ell > 0. \end{cases}$$

For a general block-circulant matrix with circulant blocks (BCCB-matrix) $\mathbf{C}_{M_v M_x}$ of order $M_v M_x$ defined by

$$\mathbf{C}_{M_v M_x} = \begin{pmatrix} \mathbf{C}_0 & \mathbf{C}_1 & \cdots & \mathbf{C}_{M_v-1} \\ \mathbf{C}_{M_v-1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{C}_1 \\ \mathbf{C}_1 & \cdots & \mathbf{C}_{M_v-1} & \mathbf{C}_0 \end{pmatrix},\ \mathbf{C}_\ell = \begin{pmatrix} C_{0,\ell} & C_{1,\ell} & \cdots & C_{M_x-1,\ell} \\ C_{M_x-1,\ell} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & C_{1,\ell} \\ C_{1,\ell} & \cdots & C_{M_x-1,\ell} & C_{0,\ell} \end{pmatrix},$$

it holds that $\mathbf{C}_{M_v M_x} = (\mathbf{F}_{M_v} \otimes \mathbf{F}_{M_x})^{-1}\mathbf{\Lambda}(\mathbf{F}_{M_v} \otimes \mathbf{F}_{M_x})$. Following the discussion in the previous section we conclude that the multiplication of a vector $\mathbf{w}$ by a BCCB-matrix can be accomplished using two $2d$ DFTs and one $2d$ IDFT. Again, the DFTs and IDFT can be computed efficiently using FFTs requiring $\mathcal{O}(M_v M_x \log_2(M_v M_x))$ arithmetic operations.

The BTTB-matrix $\mathbf{T}_{\bar{m}_v m_x}$ defined in (26) can be embedded in a BCCB-matrix $\mathbf{C}_{M_v M_x}$ where M_x is the smallest power of 2 such that $M_x \geq 2m_x - 1$ and M_v is the smallest power of 2 such that $M_v \geq 2\bar{m}_v - 1$. A multiplication of a vector $\mathbf{w}$ by the matrix $\mathbf{T}$ can then be replaced by the multiplication of the vector $\tilde{\mathbf{w}}$ by the matrix $\mathbf{C}$. Here $\tilde{\mathbf{w}}$ is defined as the extension of $\mathbf{w}$ by appending $M_x - m_x$ zeros to each block in $\mathbf{w}$ and $M_v - \bar{m}_v$ zero vectors of dimension M_x.

We summarize the computation of the integral in (23) as follows:

- Interpolate values from grid-points in (s, v) to equidistant points x_i between $x_{\min}$ and $x_{\max}$ and equidistant points v_j between 0 and $\bar{v}_{\max}$.
- Compute the BTTB-matrix $\mathbf{T}_{\bar{m}_v m_x}$ and embed this matrix in a BCCB-matrix $\mathbf{C}_{M_v M_x}$.
- Compute $\bar{\mathbf{I}}$ using the algorithm described above using FFTs.
- Interpolate $\bar{I}_{i,j}$ to the original grid in (s, v).

3.5 Numerical Experiments

In this section we price European and American *put* options under the Bates and the SVCJ models. We compare two alternative approaches to solve the discretized systems:

the algebraic multigrid (AMG) method, and the LU decomposition method. The operator splitting (OS) method [24, 27] is employed to enable the LU decomposition method for American options as well. Similarly, the projected algebraic multigrid (PAMG) method is adopted instead of the AMG for American options. These combinations lead to a total of eight different cases.

The used AMG implementation is based on the Ruge-Stüben algorithm [43] and the PAMG method is described in [51]. This algorithm generates automatically the coarse grid problems using only the left-hand side matrix. In the considered option prices problems, the dimension of the problem reduces by a factor between two and three in each coarsening of the problem. For the grids specified in Table 1, the number of levels generated by the multigrid methods varies between 7 and 16. The AMG methods employ a multigrid V-cycle. The smoother for moving downwards and upwards is one (projected) Gauss-Seidel iteration over all points, followed by one more iteration over the so-called fine-points (F-points). For the LU decompositions, we employed the UMFPACK version 5.6.1 [14]. For fast Fourier transforms, we employed FFTW version 3.3.2 [20]. We performed the experiments on a Mac laptop with 2 GHz Intel Core i7 processor and 8 Gbytes of memory.

The shared parameters and the grid refinement schedule employed in the numerical experiments are listed in Table 1. In addition, under the SVCJ model the correlation between jumps is set as $\rho_J = -0.5$, and the mean of the exponentially distributed jump sizes in variance is set as $\nu = 0.2$. The coefficients for the grid generating functions in (7) are defined by the parameters $p_K = 7/16$ and $\alpha = 2$. The truncation boundaries for the integrations are defined by $x_{\min} = \log s_2 - \frac{1}{8}(\log s_{\max} - \log s_2)$, $x_{\max} = \log s_{\max}$, and $\bar{v}_{\max} = \frac{5}{4} v_{\max}$. The number of grid points in the equidistant integration grid is twice as many in each direction compared to the computational grid. Reference prices, listed in Table 2, were computed using the (P)AMG method on a fine grid with 4097, 2049, and 513 nodes in s, v, and τ directions, respectively.

Figure 1 illustrates that the pricing errors of the (P)AMG and LU(+OS) approaches are almost identical. Option prices, ratios of consecutive errors, average iterations and CPU times for the European options under the Bates and SVCJ models are listed in Tables 3 and 5 for the AMG approach, and in Tables 4 and 6 for the LU decomposition approach. Similarly, the numerical results for American options under the Bates and SVCJ models are reported in Tables 7 and 9 for the PAMG approach, and in Tables 8 and 10 for the LU+OS approach.

The ratios of consecutive errors are computed using the l_2 norm with respect to the reference prices. The ratios are, on average, of second order. On the finest grid, in some cases the ratios are as high as 7. We also computed the ratio between the change in the solution between grids 5 to 6, and 6 to 7. These ratios were around 4 also on the finest grids, which suggests that the high ratios of consecutive errors, those listed in the tables, are due to the inaccuracy of the computed reference prices.

As the grid is refined, the LU decomposition approach is slightly faster than the (P)AMG method. In both cases, however, penny-accurate prices can obtained in a fraction of a second under the Bates model, and in about two seconds under the SVCJ model.

Parameter	Value
Brownian correlation ρ	-0.5
Risk-free interest rate r	0.03
Dividend yield q	0
Volatility of variance σ	0.25
Rate of mean reversal κ	2
Variance mean level θ	0.04
Strike price K	100
Jump arrival rate λ	0.2
Expiry time T	0.5
Jump size log-variance δ^2	0.16
Jump size log-mean γ	-0.5
Truncation boundary $s_{\max}$	$4K$
Truncation boundary $v_{\max}$	0.5

Grid #	Nodes in s	Nodes in v	Nodes in τ
1	17	9	3
2	33	17	5
3	65	33	9
4	129	65	17
5	257	129	33
6	513	257	65
7	1025	513	129

Table 1: Shared parameters employed in numerical experiments for the Bates and the SVCJ model (Left), and the grid refinement schedule (Right).

Option type	Price at 90	Price at 100	Price at 110
European put (Bates)	11.302917	6.589881	4.191455
European put (SVCJ)	11.134438	6.609162	4.342956
American put (Bates)	11.619920	6.714240	4.261583
American put (SVCJ)	11.561620	6.780527	4.442032

Table 2: Reference prices at $s = \{90, 100, 110\}$.

Grid #	Price at 90	Price at 100	Price at 110	Avg. itns.	Ratio	CPU time(s)
1	12.3849	5.0124	3.9946	2.0		0.001
2	11.3310	6.0693	4.0525	2.0	3.56	0.002
3	11.3077	6.4645	4.1630	2.1	4.20	0.016
4	11.3103	6.5589	4.1862	3.2	3.99	0.137
5	11.3029	6.5821	4.1880	2.9	3.79	0.984
6	11.3035	6.5880	4.1908	2.2	4.22	6.746
7	11.3026	6.5894	4.1914	2.8	3.72	64.584

Table 3: Average iterations, CPU times, ratios of convergence and option prices at $s = \{90, 100, 110\}$ for the European put option under the Bates model computed with AMG.

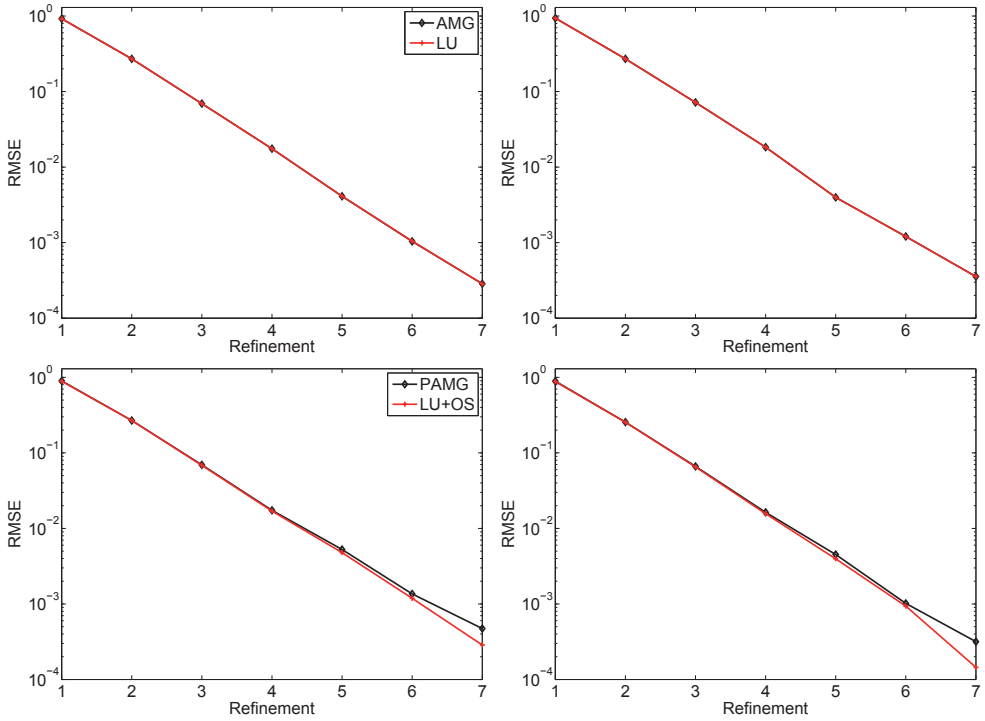

Figure 1: Absolute errors under the Bates and SVCJ models computed with the (P)AMG and LU decomposition approaches: European option under the Bates model (Top Left), European option under the SVCJ model (Top Right), American option under the Bates model (Bottom Left), and American option under the SVCJ model (Bottom Right).

Grid #	Price at 90	Price at 100	Price at 110	Avg. itns.	Ratio	CPU time(s)
1	12.3849	5.0124	3.9946	–		0.001
2	11.3310	6.0693	4.0525	–	3.56	0.003
3	11.3077	6.4645	4.1630	–	4.20	0.013
4	11.3103	6.5589	4.1862	–	3.99	0.081
5	11.3029	6.5821	4.1880	–	3.78	0.586
6	11.3034	6.5880	4.1908	–	4.17	4.556
7	11.3026	6.5894	4.1914	–	3.78	40.946

Table 4: Average iterations, CPU times, ratios of convergence and option prices at $s = \{90, 100, 110\}$ for the European put option under the Bates model computed with LU decomposition.

Grid #	Price at 90	Price at 100	Price at 110	Avg. itns.	Ratio	CPU time(s)
1	12.3827	5.1852	4.2076	2.0		0.001
2	11.2023	6.1214	4.2195	2.0	3.74	0.004
3	11.1462	6.4897	4.3191	2.1	4.15	0.039
4	11.1445	6.5810	4.3401	3.0	4.07	0.285
5	11.1359	6.6028	4.3410	2.7	4.42	2.811
6	11.1357	6.6080	4.3430	2.4	3.98	26.574
7	11.1345	6.6090	4.3432	3.0	7.00	277.622

Table 5: Average iterations, CPU times, ratios of convergence and option prices at $s = \{90, 100, 110\}$ for the European put option under the SVCJ model computed with AMG.

Grid #	Price at 90	Price at 100	Price at 110	Avg. itns.	Ratio	CPU time(s)
1	12.3827	5.1852	4.2076	–		0.002
2	11.2023	6.1214	4.2195	–	3.74	0.006
3	11.1462	6.4897	4.3190	–	4.15	0.036
4	11.1445	6.5810	4.3401	–	4.07	0.237
5	11.1359	6.6028	4.3409	–	4.41	2.598
6	11.1357	6.6080	4.3430	–	3.98	23.991
7	11.1345	6.6090	4.3432	–	7.00	246.697

Table 6: Average iterations, CPU times, ratios of convergence and option prices at $s = \{90, 100, 110\}$ for the European put option under the SVCJ model computed with LU decomposition.

Grid #	Price at 90	Price at 100	Price at 110	Avg. itns.	Ratio	CPU time(s)
1	12.7091	5.0795	4.0487	2.0		0.001
2	11.5471	6.1571	4.1120	2.0	3.40	0.003
3	11.5941	6.5718	4.2277	2.1	3.91	0.017
4	11.6189	6.6767	4.2537	3.2	3.88	0.140
5	11.6169	6.7040	4.2569	2.9	3.29	1.015
6	11.6195	6.7115	4.2605	2.3	3.90	6.778
7	11.6193	6.7135	4.2613	2.8	2.86	65.024

Table 7: Average iterations, CPU times, ratios of convergence and option prices at $s = \{90, 100, 110\}$ for the American put option under the Bates model computed with PAMG.

Grid #	Price at 90	Price at 100	Price at 110	Avg. itns.	Ratio	CPU time(s)
1	12.6810	5.0757	4.0446	–		0.001
2	11.5537	6.1576	4.1115	–	3.38	0.003
3	11.5992	6.5736	4.2282	–	3.97	0.013
4	11.6217	6.6778	4.2541	–	3.92	0.080
5	11.6187	6.7047	4.2572	–	3.52	0.592
6	11.6204	6.7119	4.2606	–	4.06	4.582
7	11.6197	6.7137	4.2614	–	4.14	41.348

Table 8: Average iterations, CPU times, ratios of convergence and option prices at $s = \{90, 100, 110\}$ for the American put option under the Bates model computed with the LU+OS approach.

Grid #	Price at 90	Price at 100	Price at 110	Avg. itns.	Ratio	CPU time(s)
1	12.8186	5.2849	4.2876	2.0		0.001
2	11.5035	6.2465	4.3055	2.0	3.54	0.005
3	11.5365	6.6405	4.4118	2.1	3.81	0.039
4	11.5616	6.7445	4.4362	3.0	3.98	0.285
5	11.5594	6.7713	4.4386	2.7	3.62	2.808
6	11.5617	6.7783	4.4415	2.4	4.44	27.276
7	11.5611	6.7800	4.4420	3.0	3.25	276.467

Table 9: Average iterations, CPU times, ratios of convergence and option prices at $s = \{90, 100, 110\}$ for the American put option under the SVCJ model computed with the PAMG.

Grid #	Price at 90	Price at 100	Price at 110	Avg. itns.	Ratio	CPU time(s)
1	12.7909	5.2869	4.2881	–		0.002
2	11.5121	6.2488	4.3060	–	3.52	0.006
3	11.5419	6.6425	4.4124	–	3.87	0.036
4	11.5658	6.7461	4.4368	–	4.07	0.239
5	11.5616	6.7722	4.4390	–	3.97	2.462
6	11.5628	6.7788	4.4417	–	4.22	24.777
7	11.5617	6.7803	4.4421	–	7.67	248.852

Table 10: Average iterations, CPU times, ratios of convergence and option prices at $s = \{90, 100, 110\}$ for the American put option under the SVCJ model computed with the LU+OS approach.

4 Conclusions

In this work we have considered the numerical pricing of options under the Bates model and the SVCJ model. For the time discretization we employed the second-order accurate IMEX-CNAB scheme which treats the differential operator implicitly and the integral term explicitly. This way we avoid having to solve systems of equations with dense matrices. The matrix-vector multiplications arising from the explicit jump terms can be computed efficiently using FFTs.

European options lead to linear systems of equations, which under the IMEX-CNAB scheme can be solved efficiently by employing AMG method or LU decomposition. For American options a common approach is to formulate the pricing problem as an LCP. Here we solve these LCPs by employing either PAMG or operator splitting in combination with LU decomposition. Both these methods were employed to price European/American options under the Bates/SVCJ model, rendering a total set-up of 8 combinations.

Numerical experiments show that the (P)AMG and LU(+OS) approaches produce almost identical prices. Since the operator splitting technique does not essentially reduce the accuracy in the solution and the LU decomposition can be precomputed prior to the time-stepping, the LU(+OS) methodology turns out to be surprisingly accurate and fast. To the best of our knowledge, this is the first paper where American options under the SVCJ model are priced.

References

[1] Y. Achdou and O. Pironneau, *Computational methods for option pricing*, vol. 30 of Frontiers in Applied Mathematics, SIAM, Philadelphia, PA, 2005.

[2] A. Almendral and C. W. Oosterlee, *Numerical valuation of options with jumps in the underlying*, Appl. Numer. Math., 53 (2005), pp. 1–18.

[3] U. Ascher, S. Ruuth, and B. Wetton, *Implicit-explicit methods for time-dependent partial differential equations*, SIAM Journal on Numerical Analysis, 32 (1995), pp. 797–823.

[4] L. V. Ballestra and C. Sgarra, *The evaluation of American options in a stochastic volatility model with jumps: an efficient finite element approach*, Comput. Math. Appl., 60 (2010), pp. 1571–1590.

[5] D. S. Bates, *Jumps and stochastic volatility: Exchange rate processes implicit Deutsche mark options*, Review Financial Stud., 9 (1996), pp. 69–107.

[6] F. Black and M. Scholes, *The pricing of options and corporate liabilities*, J. Political Economy, 81 (1973), pp. 637–654.

[7] A. Brandt and C. W. Cryer, *Multigrid algorithms for the solution of linear complementarity problems arising from free boundary problems*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 655–684.

[8] P. Carr, H. Geman, D. B. Madan, and M. Yor, *The fine structure of asset returns: an empirical investigation*, J. Business, 75 (2002), pp. 305–332.

[9] C. Chiarella, B. Kang, G. H. Meyer, and A. Ziogas, *The evaluation of American option prices under stochastic volatility and jump-diffusion dynamics using the method of lines*, Int. J. Theor. Appl. Finance, 12 (2009), pp. 393–425.

[10] N. Clarke and K. Parrott, *Multigrid for American option pricing with stochastic volatility*, Appl. Math. Finance, 6 (1999), pp. 177–195.

[11] R. Cont and P. Tankov, *Financial modelling with jump processes*, Chapman & Hall/CRC, Boca Raton, FL, 2004.

[12] R. Cont and E. Voltchkova, *A finite difference scheme for option pricing in jump diffusion and exponential Lévy models*, SIAM Numer. Anal., 43 (2005), pp. 1596–1626.

[13] C. W. Cryer, *The solution of a quadratic programming problem using systematic overrelaxation*, SIAM J. Control, 9 (1971), pp. 385–392.

[14] T. A. Davis, *Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method*, ACM Trans. Math. Software, 30 (2004), pp. 196–199.

[15] D. Duffie, J. Pan, and K. Singleton, *Transform analysis and asset pricing for affine jump-diffusions*, Econometrica, 68 (2000), pp. 1343–1376.

[16] B. Eraker, M. Johannes, and N. Polson, *The impact of jumps in volatility and returns*, J. Finance, 58 (2003), pp. 1269–1300.

[17] L. Feng and V. Linetsky, *Pricing options in jump-diffusion models: an extrapolation approach*, Oper. Res., 56 (2008), pp. 304–325.

[18] J. Fouque, G. Papanicolaou, and K. Sircar, *Derivatives in financial markets with stochastic volatility*, Cambridge University Press, 2000.

[19] J. Frank, W. Hundsdorfer, and J. Verwer, *On the stability of implicit-explicit linear multistep methods*, Appl. Numer. Math., 25 (1997), pp. 193–205.

[20] M. Frigo and S. G. Johnson, *The design and implementation of FFTW3*, in Proceedings of the IEEE, vol. 93, 2005, pp. 216–231.

[21] A. George, *Nested dissection of a regular finite element mesh*, SIAM J. Numer. Anal., 10 (1973), pp. 345–363.

[22] M. B. Giles and R. Carter, *Convergence analysis of Crank-Nicolson and Rannacher time-marching*, J. Comput. Finance, 9 (2006), pp. 89–112.

[23] S. Heston, *A closed-form solution for options with stochastic volatility with applications to bond and currency options*, Rev. Financial Stud., 6 (1993), pp. 327–343.

[24] S. Ikonen and J. Toivanen, *Operator splitting methods for American option pricing*, Appl. Math. Lett., 17 (2004), pp. 809–814.

[25] ——, *Componentwise splitting methods for pricing American options under stochastic volatility*, Int. J. Theor. Appl. Finance, 10 (2007), pp. 331–361.

[26] ——, *Efficient numerical methods for pricing American options under stochastic volatility*, Numer. Methods Partial Differential Equations, 24 (2008), pp. 104–126.

[27] ——, *Operator splitting methods for pricing American options under stochastic volatility*, Numer. Math., 113 (2009), pp. 299–324.

[28] K. Ito and J. Toivanen, *Lagrange multiplier approach with optimized finite difference stencils for pricing American options under stochastic volatility*, SIAM J. Sci. Comput., 31 (2009), pp. 2646–2664.

[29] S. G. Kou, *A jump-diffusion model for option pricing*, Management Sci., 48 (2002), pp. 1086–1101.

[30] Y. Kwon and Y. Lee, *A second-order finite difference method for option pricing under jump-diffusion models*, SIAM J. Numer. Anal., 49 (2011), pp. 2598–2617.

[31] ——, *A second-order tridiagonal method for American options under jump-diffusion models*, SIAM J. Sci. Comput., 33 (2011), pp. 1860–1872.

[32] G. Linde, J. Persson, and L. von Sydow, *A highly accurate adaptive finite difference solver for the black-scholes equation*, Int. J. Comput. Math., 86 (2009), pp. 2104–2121.

[33] P. Lötstedt, J. Persson, L. von Sydow, and J. Tysk, *Space-time adaptive finite difference method for european multi-asset options*, Computers and Mathematics with Applications, 53 (2007), pp. 1159–1180.

[34] R. C. Merton, *Theory of rational option pricing*, Bell J. Econom. Management Sci., 4 (1973), pp. 141–183.

[35] ——, *Option pricing when underlying stock returns are discontinuous*, J. Financial Econ., 3 (1976), pp. 125–144.

[36] E. Miglio and C. Sgarra, *A finite element discretization method for option pricing with the Bates model*, S$\vec{e}$MA J., (2011), pp. 23–40.

[37] C. W. Oosterlee, *On multigrid for linear complementarity problems with application to American-style options*, Electron. Trans. Numer. Anal., 15 (2003), pp. 165–185.

[38] J. Persson and L. von Sydow, *Pricing european multi-asset options using a space-time adaptive fd-method*, Comput. Vis. Sci, 10 (2007), pp. 173–183.

[39] ——, *Pricing American options using a space-time adaptive finite difference method*, Math. Comput. Simulation, 80 (2010), pp. 1922–1935.

[40] N. Rambeerich, D. Y. Tangman, M. R. Lollchund, and M. Bhuruth, *High-order computational methods for option valuation under multifactor models*, European J. Oper. Res., 224 (2013), pp. 219–226.

[41] R. Rannacher, *Finite element solution of diffusion problems with irregular data*, Numer. Math., 43 (1984), pp. 309–327.

[42] C. Reisinger and G. Wittum, *On multigrid for anisotropic equations and variational inequalities: Pricing multi-dimensional European and American options*, Comput. Vis. Sci., 7 (2004), pp. 189–197.

[43] J. W. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid methods, vol. 3 of Frontiers Appl. Math., SIAM, Philadelphia, PA, 1987, pp. 73–130.

[44] S. Salmi and J. Toivanen, *An iterative method for pricing American options under jump-diffusion models*, Appl. Numer. Math., 61 (2011), pp. 821–831.

[45] ——, *A comparison and survey of finite difference methods for pricing American options under finite activity jump-diffusion models*, Int. J. Comput. Math., 89 (2012), pp. 1112–1134.

[46] ——, *Robust and efficient IMEX schemes for option pricing under jump-diffusion models*, Submitted to journal, (2013). Available at SSRN: http://dx.doi.org/10.2139/ssrn.2247916.

[47] S. Salmi, J. Toivanen, and L. von Sydow, *Iterative methods for pricing American options under the Bates model*, in Proceedings of 2013 International Conference on Computational Science, vol. 18 of Procedia Computer Science Series, Elsevier, 2013, pp. 1136–1144.

[48] K. Stüben, *Algebraic multigrid: An introduction with applications*, in Multigrid, Academic Press Inc., San Diego, CA, 2001.

[49] D. Tavella and C. Randall, *Pricing financial instruments: The finite difference method*, John Wiley & Sons, Chichester, 2000.

[50] J. Toivanen, *A componentwise splitting method for pricing American options under the Bates model*, in Applied and numerical partial differential equations, vol. 15 of Comput. Methods Appl. Sci., Springer, New York, 2010, pp. 213–227.

[51] J. Toivanen and C. W. Oosterlee, *A projected algebraic multigrid method for linear complementarity problems*, Numer. Math. Theory Methods Appl., 5 (2012), pp. 85–98.

[52] Y.-Y. Zhang, H.-K. Pang, L. Feng, and X.-Q. Jin, *Quadratic finite element and preconditioning for options pricing in the SVCJ model*, J. Comput. Finance, (2013). To appear. Available at SSRN: http://ssrn.com/abstract=1983106.

[53] R. Zvan, P. A. Forsyth, and K. R. Vetzal, *Penalty methods for American options with stochastic volatility*, J. Comput. Appl. Math., 91 (1998), pp. 199–218.